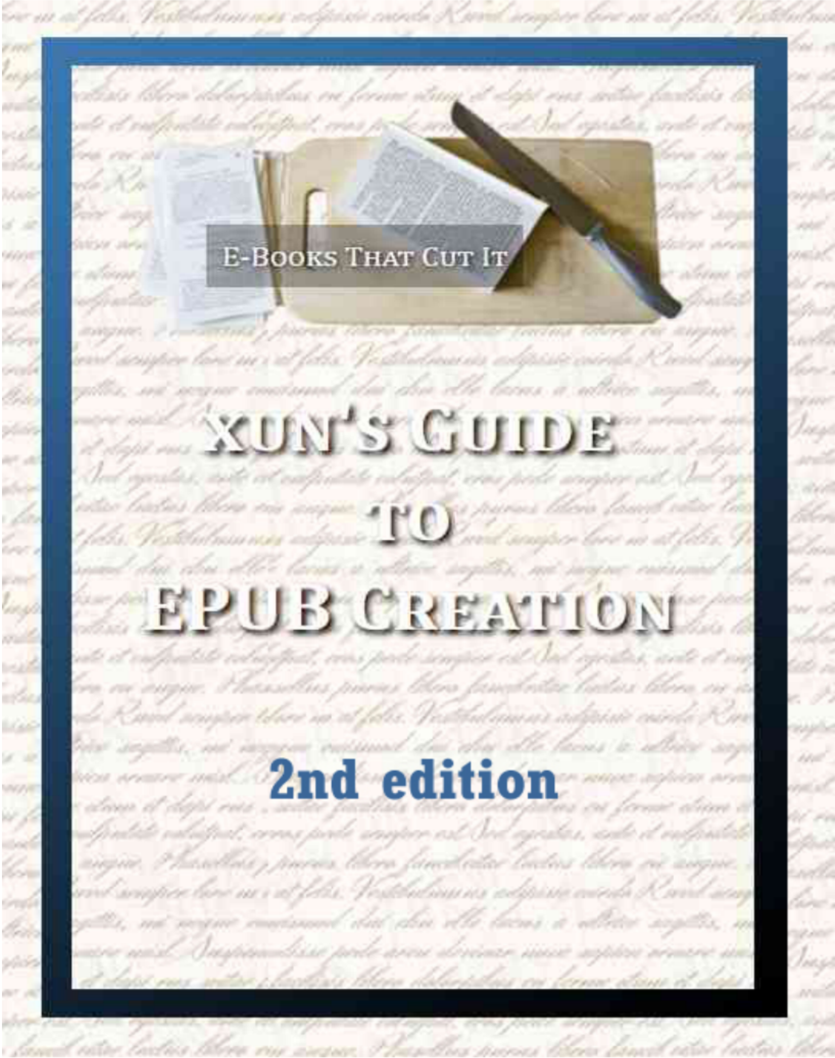


E-Books That Cut It

Xun's Guide to EPUB Creation

2nd edition

Table of Contents

- Copyleft and Thanks
- Preface
  - Book View (or PageEdit) and Code View
  - The Goal
  - The Goal of this book
  - EPUB2 vs EPUB3… and the winner is EPUB3!
  - Disclaimer
  - Finally…
- Workflow Overview
  - Step by step
    - Pre-preparation
    - Preparation
    - Create main structure
    - Design details (approximate workflow order)
    - Corrections such as:
    - Finishing touches
- Software
  - PDF software
  - Word processing
  - Image editors
  - E-book software
  - E-book reader software
- Scanning and Converting to EPUB
  - Scanning with ABBYY Transformer+
  - Converting to EPUB
  - Converting to RTF
  - Timetable or a tabbed list

Editing in RTF before converting to EPUB (Atlantis)
Creating EPUB from a web page

Style Sheets

Basic style sheet (CV)
Extended style sheet (CV)
Replacement style sheet (BV & CV)

Comments on Styling and Style Sheet Classes

Don’t style an EPUB like a web page
Don’t style an EPUB like a paper book
Classes and styles in style sheets and XHTML files: a brief explanation
How to name classes
Comment out
Uncomment
@page
.author, .title., and .publisher
body
.booklist
.copyright
.img
.firstletter
.firstline
.footnote
p
.quotefix
.ref
.smallcaps
.text

Semantic Elements and Tags

Invisible tags
Acronyms (abbreviations): the <abbr> element
Headings (<h1> etc.)
Tables
The tags <i>, <b>, <em>, <strong>, plus “italic” and “bold” in CSS

    - The <i> tag for a foreign language
    - The <cite> tag for book titles and other publications
    - Do you really have to decide between <i> and <em> (and <cite>)?
    - ARIA (Accessible Rich Internet Application)

- Working with Sigil GUI (Book View or PageEdit)
    - Introduction
        - Save a copy
        - Searching
        - Saving useful text to paste
    - Preparation
        - Copy and edit cover and images
        - Open your EPUB in Sigil
        - Remove the text in any existing style sheet
        - Search and replace all &nbsp; and &#160; with ordinary spaces
        - Remove HTML ToC in text file and generate ToC
    - Create main structure
        - Split XHTML files
        - Index
        - Split XHTML files as needed, one for each chapter
        - Add cover
        - Add XHTML files
        - More images
        - Fullpage images (copied title page, large map, etc.)
    - Design details
        - Special pages design
        - Make the chapter headings h2 or h3
        - Generate ToC again
        - Special styles such as letters and poems
        - Smallcaps

Add scenechange lines
Footnotes
Unexpected and unwanted markup when editing in Book View
Spellcheck
Smarten punctuation

Working in Sigil (Code View)

Introduction

Save a copy
Searching
Saving useful text to paste
Cleaning up code

Preparation

Copy and edit cover and images
Open your EPUB in Sigil
Add your custom CSS
Search and replace all &#160; and all &nbsp; with ordinary blank spaces
Remove HTML ToC in text file and generate ToC

Create main structure

Split XHTML files
Index
Split XHTML files as needed, one for each chapter
Add cover
Add XHTML files
More images
Fullpage images (copied title page, large map, etc.)

Design details

Special pages design
Make the chapter headings h2 or h3
Generate ToC again

        - Replace span styles with <em>, <i>, and <b>
        - Special styles such as letters and poems
        - Smallcaps
        - Add scenechange lines
        - Footnotes
        - Replace <p> with <p class="text">
        - Unexpected and unwanted markup when editing in Book View
        - Spellcheck
        - Smarten punctuation

- Corrections and Fixes
    - Using Regex mode
        - Missing chapter heading tags (BV & CV)
        - Page numbers remaining after OCR (BV & CV)
        - Bad line breaks (BV & CV)
        - Missing line breaks in conversations (BV & CV)
        - Remove incorrect line breaks due to leftover hyphens (BV & CV)
        - Remove <sup> and <sub> tags (BV & CV)
        - Remove anchors (BV & CV)
        - Remove hyperlinks (BV & CV)
        - Remove mistaken underlining (BV & CV)
        - “l” (ell) instead of exclamation mark “!” (BV & CV)
        - “P” instead of question mark “?” (BV & CV)
        - Italic or bold applied to single letter (BV & CV)
        - For a different first line following headings (CV)
        - For a different first line following scenebreaks/ scenechanges (CV)
    - Using Normal mode
        - Replace non-breaking spaces with HTML entity or numerical.
        - / (slash) instead of...
        - “1” (one) or “l” (ell) instead of “I” (BV & CV)
        - “rn” instead of “m” and vice versa (BV & CV)

- "lie" instead of "he" (BV & CV)
- "Pie" or "Fie" instead of "He" or "She" (BV & CV)
- Unwanted span styles not previously corrected (CV)
- Apply quotefix (CV)

The exception: Flatten ToC with Calibre e-book editor (BV & CV)

The Finishing Touches

- Add your custom CSS (BV)
- Clean up the style sheet (BV & CV)
- Add landmarks (CV)
- Add title to <head>
- Add metadata if needed
- Validate with EPUBCheck (BV & CV)
- Run Access-Aide (CV)
- Proofread (BV & CV)
- Check with e-reader (BV & CV)

Appendix 1: Searches for Easy Copying
Appendix 2: Various Recognized EPUB Types
Appendix 3: Table of Basic HTML Entities and Codes

Cover
Start of Book
Appendix 1: Searches for Easy Copying

Copyleft and Thanks

“From all of me to all of you…”

Many thanks to **dearleuk**, who undertook to read early versions of the book and made plenty of helpful comments, all incorporated in the final product.

This is an original EPUB written by xun in December 2019-January 2020.

2nd edition restructured and updated with corrections, additional information, and adaptions to EPUB3 in November 2024. And converted to EPUB3…

Version 2 of the 2nd edition in January 2025: some minor corrections, and some added musings on not styling EPUBs like web pages or paper books.

Version 2.5 of the 2nd edition in May 2025: just fixed some annoying typos, and decided to stick to “style sheet” rather than “stylesheet” in the running text.

All text and all images (screenshots and cover) created by xun.

Preface

EPUB creation is, if not an art, at least a craft. Anybody can load a PDF file into Calibre, click to have it converted to EPUB, and thus be the proud owner of a PDF-to-EPUB conversion.

However, the result is not likely to be something to be proud of. It is likely to resemble a train wreck. Lines that are broken into new paragraphs in the middle of sentences, page numbers left in the middle of the new pages, blocking the reading flow, OCR (Optical Character Recognition) mistakes that make paragraphs and whole pages impossible to read… This goes for every quick and easy PDF to EPUB converter I have ever seen.

> *This is also why I would recommend you to download PDF when possible from sources of free and/or public domain books, and then create your own EPUB. Often (but not always, do check!), the EPUBs on offer are precisely like that: automatically converted, but not quality proofed, and so almost unreadable.*

If you want to create a really good EPUB from a PDF file, you will have to be prepared to spend some time doing it, and you need to learn to use the tools for doing it properly. It is worth it. You can be truly proud of the result when your work is closing in on perfection…

In other words: use **Sigil**.

Book View (or PageEdit) and Code View

I almost always work in Code View, since I like being in full control of the result, creating style classes as needed. But you can make decent EPUBs without “coding”, too. It is very much better to design in Book View than just stopping after the conversion from PDF to EPUB; that is likely to be a book that is difficult to read due to lots of OCR mistakes, lack of structure and with inconsistent formatting.

When I say Book View, I refer equally to PageEdit, or other GUI editors that can be used with Sigil. I personally prefer to be able to shift back and forth between the two views while having the focus (selection) stay in place, so I use the last Sigil version to have Book View, which fortunately supports EPUB3. I switch to a later version for the very last steps: running the EPUBCheck and Access-Aide plugins, and the proofreading.

On a side note, you can also open the book in the Calibre e-book viewer while working in Code View, in order to see the result of applied styles, etc.

Working in Book View is faster and easier than being perfectionistic in Code View. That suggests another occasion when Book View may be the better alternative: downloaded books that look awful often have horrid coding and style sheets under the hood. Instead of looking at the mess and despair, just edit so that the books look OK in Book View!

Since some steps are the same, and other steps are done differently, I have created separate chapters on working in Book View and working in Code View. That way, it should be easy to follow the detailed guide without having to skip text that doesn't apply to your chosen method. This, however, also means that some text is the same in those two chapters, since some steps are indeed the same. If you read both chapters, there will be some *déjà vu* moments.

The Goal

What do you want from an e-book?

You want it to be **legible, easy to read**, so there should no (or a minimum of) misspellings or OCR mistakes turning words into rubbish, and no bad line breaks.

You want to be **able to navigate** the book, so there should be headings and a Table of Content that links to the different parts of the book.

You want **indications when the scene changes**, by a blank line or some special characters, alerting you to move elsewhere in the perspective, space or time of the book world.

You want to be able to **tell utterances from running text**, so there should be quotation marks, correctly and consistently applied, making it clear who says what.

You want to be able to **tell songs, poems, letter, telegrams, and quotations from running text and ordinary utterances**, so these should be formatted somewhat differently (often by a larger left margin and italics).

Your EPUB design should meet those expectations.

I would say that it is also **desirable that the EPUB is designed with different preferences and needs in mind.** Many e-readers allow the reader to choose fonts, adjust font-size and line-height, and even font-weight, and to set left and right margins. Don't block those choices by creating a style sheet that forces the reader to use certain fonts, line-heights, etc.

To be sure, it is not only a matter of respecting others' preferences: e-readers are saviours for many people with poor eye-sight, exactly because they can adjust font-sizes, font-weights, etc. and keep reading when paper books are no longer a possibility. Don't block it by CSS!

It is also important to **make use of semantic elements and tags in order to make the EPUB more accessible** to different kinds of readers. This mainly concerns those who create EPUBs in Code View since many features require editing the code.

In addition, I usually strive to create as clean markup as possible, and a style sheet that is practically self-explanatory, so that anybody who wants to edit one of "my" books will find it easy to do so. My classes are called things like "text" and "letter" rather than "t1" and "block03", etc.

This addition has little to do with the above expectations, though. An EPUB can have horribly confusing markup and unintelligible CSS, and still meet the basic expectations — though accessibility is likely to suffer.

In other words, as long as you meet the reasonable expectations above, you're fine. Clean markup and clear CSS are nice, but bonuses.

The Goal of this book

Ironically, this EPUB is designed to be read at a computer, preferably a desktop computer with a decently large monitor. It doesn't look that good on a small e-reader. This is a guide, not entertainment.

Open the book in Calibre's e-reader on your computer and follow the steps while you work with your book. Select, copy, and paste text from the book as needed.

Using Calibre's e-reader has the additional advantage of showing the levelled ToC (Table of Contents) correctly: some e-readers will flatten ToC automatically and show all the little subheadings along with the major headings. Considering the fact that some text is duplicated in the "Working in Sigil (Code View)" and the "Working with Sigil GUI (Book View or PageEdit)" chapters, that provides plenty of room for confusion.

On a side note: in my opinion, the Calibre e-reader's default font size is a bit largish and so needlessly reduces the amount of text on each page. If you agree, go to Preferences - Fonts, and change it.

EPUB2 vs EPUB3… and the winner is EPUB3!

I still do EPUB2, mainly because I usually create EPUBs from my own scans, which are then converted to EPUB using an old version of ABBYY which will “only” convert to EPUB2, not EPUB3. (On the bright side, I bought a license for ABBY and can thus use the program “forever”, instead of paying for a subscription that has to be continously renewed, which is how later versions are distributed.)

There are many advantages with EPUB3, but most of them do not matter in the creation of the kind of books (novels) I usually convert. I have noticed that even with EPUB2 (HTML4 with CSS2), there are many tags/ classes that do not work as they should in every reader. I also know that with HTML5 (EPUB3) we are back to having to check what works in various web browsers, so I suspect that it is quite difficult to predict how tags will work in various readers. Simple, very simple, HTML4/EPUB2 should do. For starters, when creating the EPUB…

However:

With EPUB3 becoming the standard, and with the added accessibility features in EPUB3, I now take the further step after my EPUB2 books are created. I convert to EPUB3 using the ePub3-itizer plugin. I add semantic tags to the XHTM files, and then run the Access-Aide plugin to make sure that the books work as well as possible for all readers. Decent EPUB2 books thus become decent EPUB3 books. I make sure to keep the EPUB3 working with EPUB2 readers by keeping the *toc.ncx* file.

The work flow described here can be used if you first create an EPUB2 and convert it to EPUB3, but it will work just as well if you make an EPUB3 from the start. The difference will be that you won’t convert your EPUB3.

Disclaimer

This book was mainly written (with a lot of text copied from my old EPUB web page and some of my forum posts at various sites) using an old version of Sigil, 0.9.14, installed on an offline Windows machine. I have checked against later versions, but may have missed some changed settings and features, and what’s more: these may change in future releases.

In other words, I cannot guarantee that you will not have to look for some settings, similar to the ones I mention, elsewhere in the programs — now or in the future.

Finally…

…this is all about novels, with few or no images and no need for specific (fixed) layout of pages. If you scan a textbook it may well be best to leave it in PDF, possibly using ScanTailor and other software to fix imperfections. A different kind of work, that.

Workflow Overview

These are brief notes of my workflow when converting books from PDF to EPUB. Some entries are specific for Book View (**BV**) or Code View (**CV**).

The steps are described in more detail in the chapters “Working with Sigil GUI (Book View or Page Edit)” and “Working in Sigil (Code View)”, respectively.

Step by step

Pre-preparation

- Run OCR if needed for the original PDF file, making it searchable. Keep it open for quick reference while working on the EPUB.
- Open your EPUB in Sigil. If your project is not already EPUB3, run the ePub3-itizer plugin to make it so.

Preparation

1. Copy and edit cover and images
2. **(BV)** Remove text in existing style sheet (N.B. not italics) / **(CV)** Add custom CSS (N.B. check italics first)
3. Search and replace all &nbsp; and &#160; with ordinary blank spaces
4. Remove HTML ToC in file and generate ToC

Now zoom out the PDF for easy scrolling and scroll through the PDF and EPUB side-by-side as it were.

Create main structure

1. Split XHTML files for front and back of the book
2. Index (remove)
3. Split XHTML files as needed, one for each chapter
4. Add cover
5. Add XHTML files if needed
6. More images
7. Fullpage images (copied title page, large map, etc.)

Design details (approximate workflow order)

1. Special pages design (titlepage, copyright page, etc.) (**CV**: and add EPUB types plus sections)
2. Make the chapter headings h2 or h3 (**CV**: and add EPUB types plus sections)
3. Generate ToC again
4. **(CV)** Replace span styles with <em>/<i> and <b>
5. Special styles such as letters and poems
6. Smallcaps (what to do)
7. Add scenechange lines
8. Footnotes (add if needed)
9. **(CV)** Replace <p> with <p class="text">
10. Unexpected and unwanted markup when editing in Book View
11. Spellcheck
12. Smarten punctuation

Corrections such as:

- Remove bad line breaks
- Add “firstlines”
- Fix various OCR mistakes

Finishing touches

1. **(BV optional)** add custom CSS
2. Clean up style sheet
3. **(CV)** add landmarks to nav file
4. Add title to <head>
5. Add metadata if needed
6. Validate with EPUBCheck
7. **(CV)** Run AccessAide
8. Proofread
9. Check in different e-readers

Software

For all that I am otherwise on Linux and the occasional Mac, I use Windows for creating EPUBs, in order to have access to my purchased and excellent software. You can run Sigil on Linux and Mac, too, and you can use GIMP or other software for editing images instead of Adobe Photoshop. However, as far as I know, there is nothing as good as ABBYY (and the back-up Nuance Power PDF) for scanning and OCRing PDFs on Linux or Mac.

There is of course a lot of software out there. This is just a catalogue of programs that are well-known in EPUB-making circles, and that I personally have found useful.

I don’t provide links to software here (apart from the threads for Sigil plugins at the *MobileRead* forum); URLs change. Search the net for those that interest you, and make sure that you download from homepages only. That way there is little risk that your download will contain malware.

PDF software

Adobe Acrobat Pro, the classic for Mac and Windows, is fine as long as you have nice, clean PDF files to OCR. If there is a coloured background or somewhat blurry text, Acrobat is not that good in my experience. When I started out, using Acrobat, I used to clean imperfect PDF files as images in Photoshop and then re-assemble them. You will save time using better OCR software if you work with old books and/or imperfect scans. Acrobat XI was the last version for which you could buy a license for continued use of that version; from 2015 onwards, it is part of the “Document Cloud”.

ABBYY (Mac and Windows) is much loved by e-book makers for its great OCR capacities. I use the PDF Transformer+ since I like to be able to do a bit more than just OCR. There is also FineReader, that I believe has a few more options, but use the same OCR engine as PDF Transformer+. ABBYY doesn’t come cheap, and it is now something to subscribe to, rather than buying a license for, Oh well.

Nuance Power PDF (Mac and Windows). An alternative to ABBYY (which has apparently been bought by a different company and renamed to **Kofax Power PDF**). They can both cope with PDFs in which the text from the reverse side shines though and the background is brownish. Sometimes one will perform a decent OCR when the other fails. Power PDF uses the same OCR engine as the renowned OmniPage: if you are just going to OCR, or if you work with textbooks rather than novels, you might want to use OmniPage instead, though I have (1) found that I do not need those advanced processing modes, and (2) I like to have a PDF program so that I can convert image PDFs to searchable PDFs.

There is one big annoyance with Power PDF, though: it comes with a poorly integrated Software Manager causing nag screens about updates to appear, and you simply cannot turn them off even if you do not want that particular update. In order to get rid of the software manager, you have to download an uninstaller and also (before you start Power PDF again) remove the FlexNet folder that resides in the hidden Program Data folder.

Like ABBYY, Power PDF is relatively expensive, but is on sale now and then. And you buy a license, not a subscription.

Scan Tailor is free software for editing scans (images); it can be downloaded for Windows and is available in many Linux repositories. For Mac and for Linux distros lacking Scan Tailor in the repos, there is source code to compile. It can be used for fixing orientation, split two-page scans into single pages, and to deskew pages that were skewed during scanning — which often happens when scanning an open book, and tends to cause many OCR mistakes. There is a very good user guide that helps you through the stages of processing the images that will later become a fine-looking PDF.

Briss is a freeware tool for Linux, Mac, and Windows that also allows you to split two-page scans/PDFs. I don’t recommend trying to make an EPUB out of anything but single page PDFs.

Word processing

Atlantis is a word processor for Windows. It looks like an older MS Word version but it is really quite powerful, and has the additional advantage, compared to MS Word, that you can export your documents to EPUB format. It is not freeware, but the license (to be paid after the 30 day trial period) isn’t that expensive, either.

LibreOffice is a freeware Office suite with Linux, Mac and Windows versions. Documents created or edited in Writer can be exported to EPUB2 or EPUB3, and you can even choose fixed layout rather than the native reflowable one. I would say that content that requires fixed layout should be exported to PDF instead...

The resulting markup of the XHTM files is not a pretty sight, though. Personally, I would rather save a document to PDF and then use ABBYY for the EPUB conversion. If you do use LibreOffice (Writer) and consider cleaning up before exporting to EPUB, **do not use the Format - Clear Direct Formatting option**. It removes the all-important italics!

Image editors

Adobe Photoshop (for Mac and Windows) is the classic, and very useful once you learn how to use it. Since 2013 it's subscription software ("Creative Cloud") rather than a license for a version, which means that you can buy the right to use it for a short period of time, but on the other hand you have to keep paying to keep using it.

GIMP, GNU Image Manipulation Program, have many of the Photoshop features, and is free software for Linux, Mac and Windows. Like Photoshop there is a bit of a learning curve, but it is great when you know how to use it.

IrfanView, freeware for Windows is excellent for changing (reducing) the size of an image. You can do some edits in there, too, but the GUI is far from intuitive.

paint.net (note that this is the name of the program, while the URL is getpaint.net) is freeware for Windows. It is probably more intuitive to use if you want a GUI that is easier to start using right away, compared to Photoshop and GIMP, but also not as rich in features.

E-book software

Sigil for creating and editing EPUBs. I almost only work in the very useful Code View, but it is very nice to be able to quickly change into 'Book View' and see what your code looks like when it has been parsed. Sigil also contains tools that, for example, allows you to create and edit the Table of Contents and check that your EPUB complies with the standard. In addition there is spellchecking (you may need to "train" it) and both normal and Regex search-and-replace, which is really a must. Sigil is downloadable freeware for Mac and Windows, and later versions are included in many Linux repositories. It can also be built from source.

> ***N.B. Since version 0.9.2, you cannot "prettify" and tidy HTML code without having it automatically mended.*** *This is not a problem if you "only" work in Book View (since Sigil is unlikely to introduce markup errors that way), but, unless the mending algorithm is perfect (which is unlikely), automatically mending custom HTML code may lead to unwanted changes. Sigil 0.9.2 onwards also mends automatically when you save the EPUB (File - Save or Ctrl+S), but there is still an error message if you try to change from Code View to Book View and there's an error in the markup. Previously (assuming settings like the ones detailed below), there was an error message when trying to save the problem line was somewhat indicated when*

trying to switch to Book View ("an error on line 160 or above"). From 0.9.2 the page will be rendered in Book View up to the error, with a message saying so, so you can go back to Code View and fix it without having it automatically mended.

In Sigil 0.9.1 and earlier, you can set your preferences so that your HTML code will be "prettified" but not automatically mended. *Set Preferences → "Clear Source": "Use Pretty Print Tidy", which will make your HTML code easy to survey. Do not use "HTML Tidy", since this might perform corrections to your code that may prove fatal for your project, depending on the perceived problem. Check "Automatically Clean and Format HTML Source Code" for "Save". If there is something syntactically wrong with your code, you well get an error message offering to take care of the problem for you as you save your e-book. I always refuse the offer in order to go find and sort out the problem myself. That way I know the corrections made are the ones I want.*

N.B. Since version 0.9.15, Sigil no longer has the switch to Book View feature, *but there is a preview for checking what edits will look like when reading the book. The buttons for quick formatting are still there, but you see the resulting code, not what it looks like in a book.*

Instead of the old Book View GUI, there is now a separate program, ***PageEdit****, for making GUI edits. It can be launched via Sigil, or used on its own for editing XHTML files; you can't open an EPUB in PageEdit on its own, but again, it can be launched via Sigil, and you can save changes you make in PageEdit and then continue in Sigil (where you need to save the changes again)*

If you prefer Sigil with the old built-in Book View, you can still download an old version (0.9.14 or older) and keep using that. That is what I do, up until it is time to validate and proofread. Why? Because with the integrated Book View, switching between Code View and Book View keeps the focus (cursor/selection) in the same place, which is sometimes very useful. Not least in a book like this, where I use Book View for pasting code snippets that would otherwise be changed into HTML code by Sigil in Code View.

There are many plugins for Sigil. These four are the most essential ones in my opinion:

- **PunctuationSmarten Sigil plugin** - turns straight quotes into curly ones, which makes for easier reading of novels (though it is not a good idea to run it on books like this one, where there are a lot of straight quotes in markup examples). See https://www.mobileread.com/forums/showthread.php?t=247088&highlight=smartenpunctuation

- **ePub3-itizer** - epub3 output plugin for Sigil, to use if you create EPUB2 and so need to convert your book to EPUB3. See https://www.mobileread.com/forums/showthread.php?t=250566

- **Access-Aide** - help improve epub accessibility - note that you need to add markup like sections and EPUB types before running the plugin, but after converting to EPUB3. See https://www.mobileread.com/forums/showthread.php?t=294900

- **EPUBCheck** - a W3C project for making sure that an EPUB (both 2 and 3 supported) meets the standards. It is integrated into Sigil by means of a wrapper, see https://www.mobileread.com/forums/showthread.php?t=248186 for the wrapper that makes W3C's EPUBCheck work from inside Sigil. Note that you cannot run the Sigil plugin on an offline computer, since it it written so that it *must* check for updates (and, possibly, the EPUBCheck Java files, even if you have downloaded them and placed them in the correct folder as per the instructions). You can, however, run EPUBCheck itself from the command line.

> ***MobileRead forums** is where to discuss the making of e-books, and meet the developers for Sigil and Calibre among others. Lots of plugins are developed and maintained there, so it's worth a look if you feel the need of expanded features.*

Calibre is a well-known freeware e-book manager for Linux, Mac and Windows that allows you to add tags and other meta data to your books, and it offers a useful overview once you have the proper tags in place.

Calibre is fine for converting between the common reflowable text formats AZW, MOBI, and EPUB. You can even use it for creating acceptable (though not really great) PDFs from EPUB. It's converting from the fixed format PDF to the reflowable EPUB (or AZW or MOBI) that will end with disaster with Calibre and other automatic converters.

While Calibre usually converts nicely from EPUB to MOBI, it will occasionally fail. Being lazy, I haven't investigated possible reasons and/or fixes for this. Instead, I open the EPUB in **Kindle Previewer** (version 1.17.1, freely downloadable from Amazon). This may take a while because it converts while opening, but when it's done you can export the EPUB as a MOBI that will usually look and behave better than a Calibre conversion. If you prefer AZW, the MOBI can then be converted in Calibre.

Calibre comes with an editor as well. I haven't used it much (just for quick fixes), always returning to Sigil - unless the need arises to flatten a ToC that refuses to flatten when it should.

Calibre can also be useful if the EPUB file size of a downloaded EPUB is way larger than reasonable due to large images. Do make a copy first in order to keep the original image quality, then open the EPUB in the Calibre editor. Click Tools - Compress images losslessly. This will actually allow you to perform lossy compression of JPEG (JPG) and WEBP images. Toy with the image quality setting; lower numbers will decrease both quality and size more. Use the Save as a copy option in order to compare with the original.

E-book reader software

When creating EPUBs, it is always good to try and check that your books will work on other devices than the one you currently use. Here are some alternatives for use on your computer.

Adobe Digital Editions can be run on Mac and Windows, and is often required for reading library e-books (and is thus supported by many e-book readers, too).

Sumatra PDF (Windows) can actually handle plenty of formats, including EPUB, and runs with a small footprint.

Bookworm and **Foliate** are both available via Flatpak for Linux (and possibly in some repos, too).

If you use **Calibre**, there’s an e-reader for checking books that can be run on its own, too. It opens all the common formats.

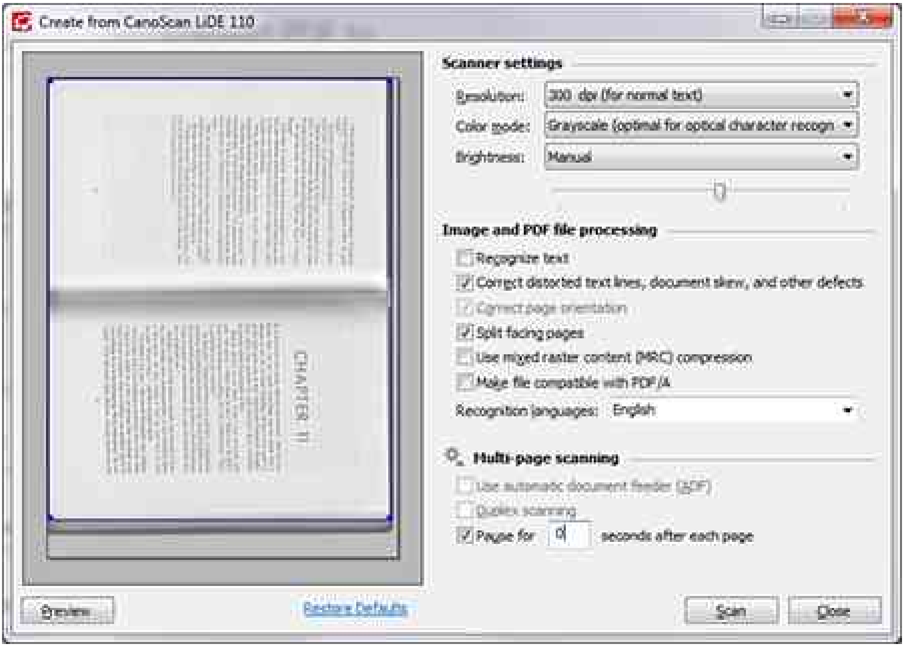
If you convert to AZW or MOBI but don’t have a Kindle reader, you can check the result in **Kindle4PC** or **Kindle4Mac**, both free to download from Amazon.

Scanning and Converting to EPUB

Scanning with ABBYY Transformer+

I have a plain little flatbed scanner, and it works well enough for the relatively few books I scan. I scan from ABBY.

These are the settings I usually use; the zero amount of time for turning pages is not so small as it may seem, since the scanner takes its time returning to original position and "warming up" before starting over.



Scanner settings

Resolution: 300 dpi (for normal text)
Color mode: Grayscale (optimal for optical character recogn
Brightness: Manual

Image and PDF file processing

☐ Recognize text
☑ Correct distorted text lines, document skew, and other defects
☑ Correct page orientation
☑ Split facing pages
☐ Use mixed raster content (MRC) compression
☐ Make file compatible with PDF/A
Recognition languages: English

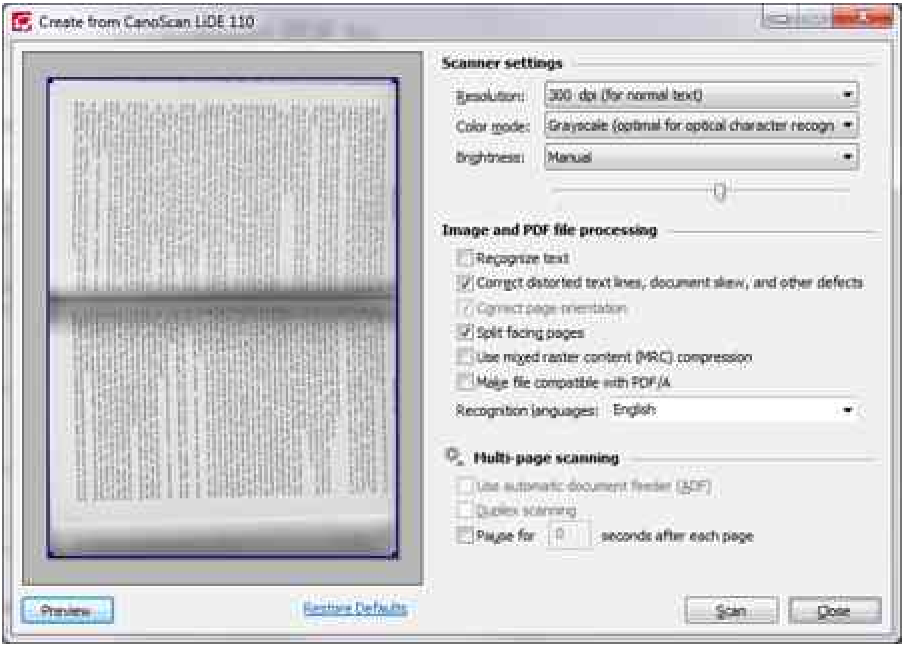
Multi-page scanning

☐ Use automatic document feeder (ADF)
☐ Duplex scanning
☑ Pause for 0 seconds after each page

Here it doesn't matter if I turn text recognition on or off while scanning, but I would like to point out that the formatting is lost if I use the similar-looking "OCR" option with the original scanner software. You do *not* want to lose italics when scanning novels.

You can cancel scanning at any time, and continue with the same document when convenient, and also re-scan any page that for some reason failed to live up to quality expectations. (This is likely to happen if you happen to move the book during the scan.) ABBYY will allow you to choose where to insert new pages.

If possible, I take care not to harm the paper books while scanning. However, some books are fairly hopeless cases. In a case like the one below, you will either have to spend a lot of time correcting misinterpreted words at the start and finish of each line, or slaughter the poor book so that you can scan the pages one by one.



Create from CanoScan LiDE 110

Scanner settings

Resolution: 300 dpi (for normal text)

Color mode: Grayscale (optimal for optical character recogn

Brightness: Manual

Image and PDF file processing

□ Recognize text

☑ Correct distorted text lines, document skew, and other defects

☑ Correct page orientation

☑ Split facing pages

□ Use mixed raster content (MRC) compression

□ Make file compatible with PDF/A

Recognition languages: English

Multi-page scanning

□ Use automatic document feeder (ADF)

□ Duplex scanning

□ Pause for 0 seconds after each page

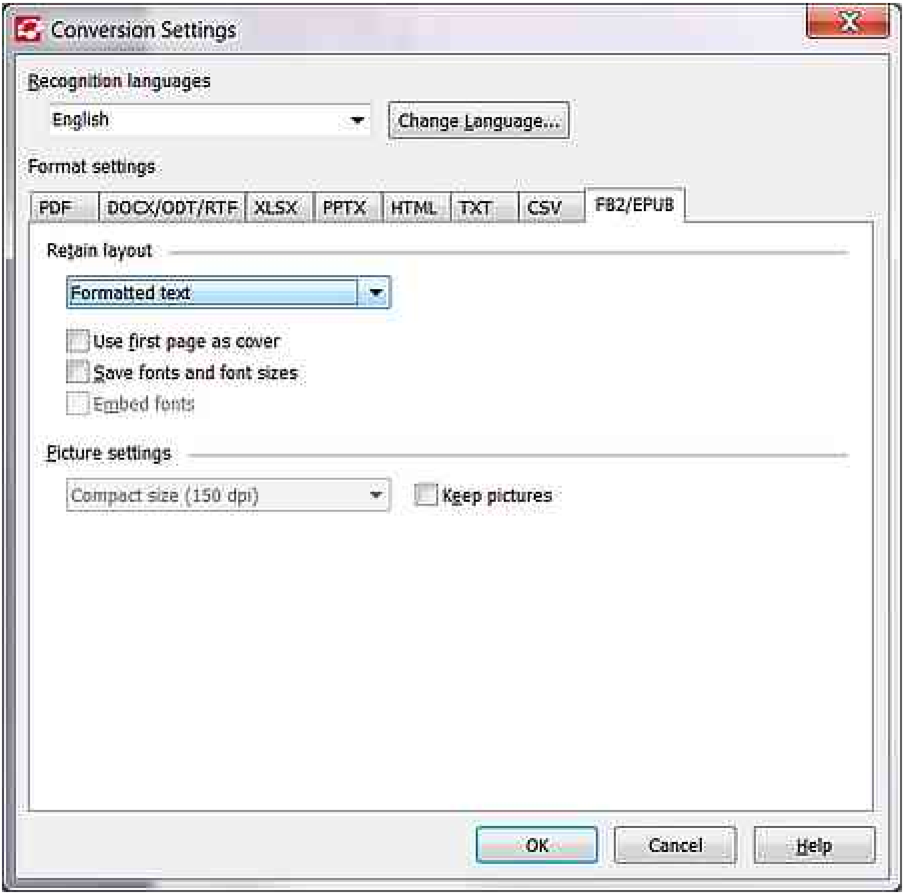
Converting to EPUB

These are my recommended settings for converting from ABBYY to EPUB. (The settings are found in the Convert To menu.)

Selecting “Formatted text” is important. If you select “Plain text” the italics will disappear!

Since ABBYY doesn’t handle images all that well, I don’t even try to make it use the first page as cover, even if there is a cover in the PDF file. It will be added later on.

I also don’t recommend saving fonts and font sizes (that gives you a complicated markup that just takes time to clear up), and absolutely not to embed fonts. Embedded fonts make the EPUB unnecessarily large, and impossible to customize for the reader to boot.



Conversion Settings

Recognition languages: English

Format settings

Retain layout: Formatted text

□ Use first page as cover

□ Save fonts and font sizes

□ Embed fonts

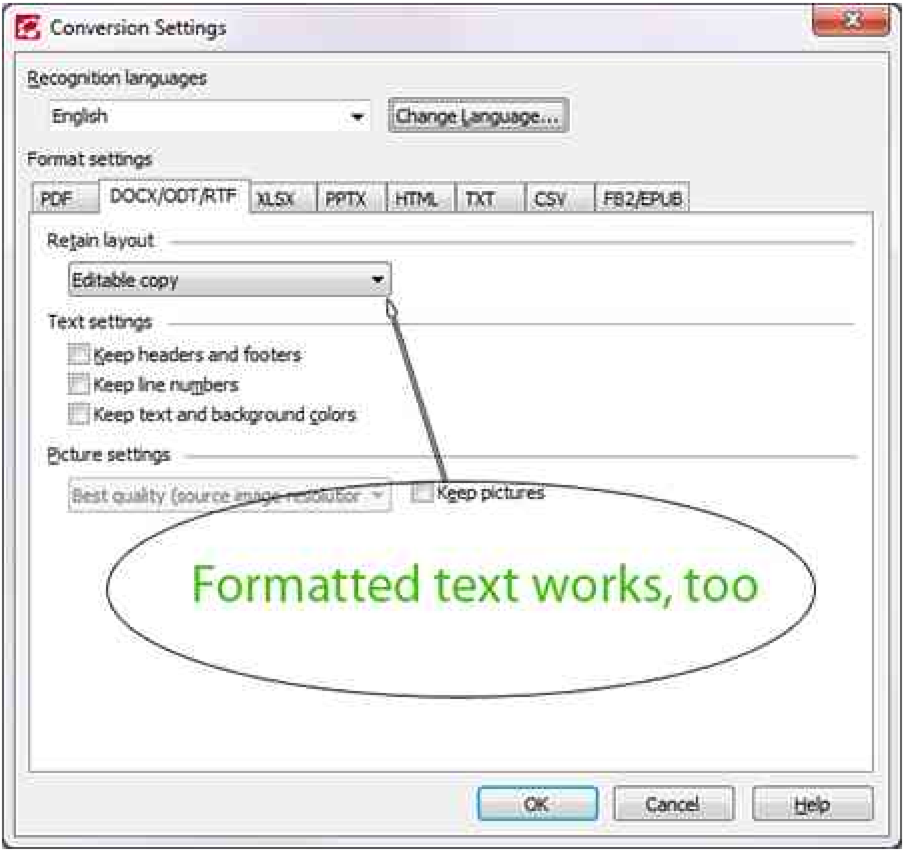
Picture settings: Compact size (150 dpi)

□ Keep pictures

Converting to RTF

If there is a lot of reconstructive work to be performed on a bad scan, I will occasionally choose to convert to RTF (Rich Text Format) instead, since I find it easier to re-write lots of text in a real word processor.

These are the settings I use in those rare cases:



Recognition languages: English

Format settings

Retain layout: Editable copy

Text settings

□ Keep headers and footers

□ Keep line numbers

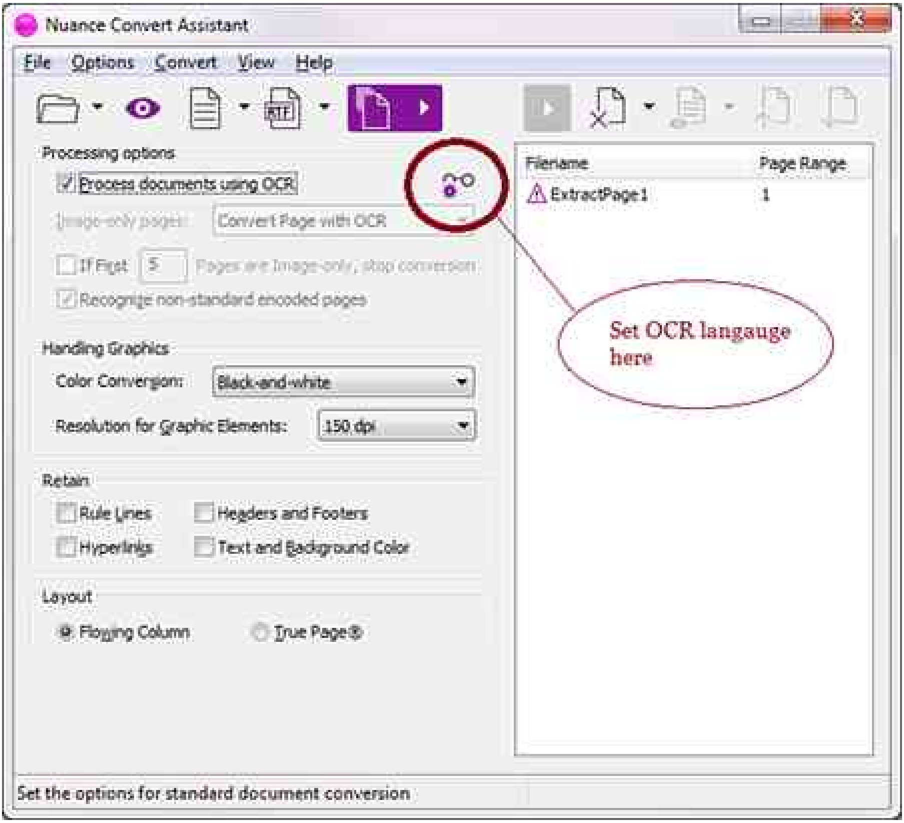
□ Keep text and background colors

Picture settings: Best quality (source image resolution) □ Keep pictures

Formatted text works, too

Timetable or a tabbed list

Mysteries, especially, sometimes contain timetables or tabbed lists, illustrating the detective's attempt to put happenings in order. If any of these have turned out very messy with ABBYY, and you have access to Power PDF, it may be a good idea to convert those pages using it; my old version can't convert directly to EPUB, but I save to RTF and then convert from Atlantis to EPUB (a separate one, and then I copy from that one to the book I'm working on). If Nuance Power PDF has managed better, as it often does, it still saves a lot of work.



Processing options

☑ Process documents using OCR

Image-only pages: Convert Page with OCR

☐ If First 5 Pages are Image-only, stop conversion

☑ Recognize non-standard encoded pages

Handling Graphics

Color Conversion: Black-and-white

Resolution for Graphic Elements: 150 dpi

Retain

☐ Rule Lines ☐ Headers and Footers

☐ Hyperlinks ☐ Text and Background Color

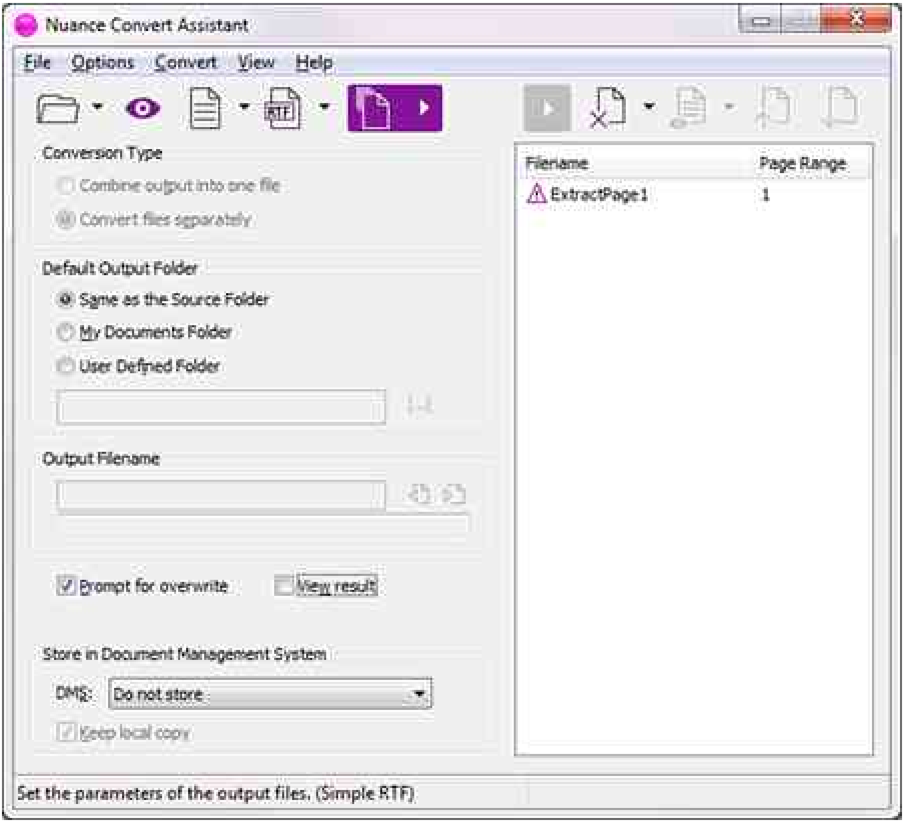
Layout

● Flowing Column ○ True Page®

Filename	Page Range
⚠ ExtractPage1	1

Set the options for standard document conversion

Set OCR langauge here



Conversion Type

○ Combine output into one file

● Convert files separately

Default Output Folder

● Same as the Source Folder

○ My Documents Folder

○ User Defined Folder

Output Filename

☑ Prompt for overwrite □ View result

Store in Document Management System

DMS: Do not store

☑ Keep local copy

Filename	Page Range
⚠ ExtractPage1	1

Set the parameters of the output files. (Simple RTF)

Editing in RTF before converting to EPUB (Atlantis)

Occasionally, a scan will be so poor that OCR will fail dramatically. This typically happens when you scan a book you can't press open enough, and not cut into separate pages.

If it is just a matter of skewed lines, it is sometimes possible to make the PDF much better using Scan Tailor. But sometimes the text near the middle will be smudged due to its distance to the scanner, and that is difficult or impossible to recover by processing the PDF.

It's a matter of taste, but in such cases, when there is a lot of text reconstruction to do, I prefer to do this in a word processor. So, in such rare cases, I don't convert from ABBYY to EPUB, but instead to RTF.

Never convert to TXT ! There are still web pages out there that recommend using text files for making EPUBs. It is madness. A plain text file will lose all formatting, notably the all-important italicized words. Italics convey meaning. Don't ever lose them!

So, RTF it is: Rich Text Formatting.

I open the RTF for editing in Atlantis in these instances, and export to EPUB when done.

Exporting a document with all formatting intact will result in rather horrid markup, though. I clean out practically everything (except italics!) before exporting to EPUB. That way there are still some, but not that many, annoyances to clear away before the real work with the EPUB can begin.

Again, this is how I do it in Atlantis (and an old version to boot). You may be able to device a similar clean-up routine in Libre Office Writer or some other word processor that allows exporting to EPUB. Just **make sure that you never lose the italics** — so don't use the "Clear Direct Formatting" option in Libre Office Writer.

Autocorrect Settings

Before you open the file with Atlantis, check the autocorrect settings (Tools → Autocorrect Options). I uncheck most of them, but highly recommend the "smart" (curly) quotes setting. Make sure that "Capitalize first letter of sentences" is unchecked — if it is checked, there will be upper case letters in the middle of sentences that were broken by section breaks.

Autocorrect

After opening the document, run "Autocorrect…". It will give you suggestions for what to correct, so you can reject anything that doesn't seem like a good idea. It will only do one kind of correction at the time, so you will need to run it again to get all corrections done. With my settings, it only needs to be run once or twice.

Spellchecking

Next; spellchecking, which you might do here and/or later on in Sigil.

Spellchecker options are found at Tools → Options → Spellcheck tab.

You set the language in Spellcheckers & Dictionaries; if you work with both British and American English books, it's a good idea to first check that you are using the right dictionary.

Check:

- Spellcheck as you type
- Underline misspellings
- Ignore Internet and file addresses

And leave everything else unchecked.

Next, run the spellchecker. Open the searchable PDF and run it alongside Atlantis as you spellcheck. (I usually use Power PDF for this; I prefer the search feature that “only” finds and selects the next occurrence.) If you are unsure about a word that the spellchecker “thinks” is wrong it is very useful to search for it in the original PDF. If it is such a common word that you risk getting too many results, search for some less common word that appears near the one you need to look up.

You may have to cancel the spellchecking and perform a manual correction, for example if a word has been incorrectly hyphenated, like “ob-“ than continues on the next line with “serve”. (The spellchecker may suggest replacements but offers no way to remove the hyphen.) Start the spellchecker where you are done by hitting the F7 key.

How to handle actual misspellings in the original text? Well, that is up to your conscience and your definition of “perfect”…

Remove Formatting

The last step before converting to EPUB is to remove as much unnecessary formatting as possible. I basically only keep italics (important!) and smallcaps (if applicable). Everything else will be restored by HTML and CSS later on; for now, I want a nice clean CSS to start with.

Atlantis tends to create different classes in the style sheet for very minor perceived differences in the text, and also creates rather horrid HTML markup in which <span> tags, inline elements, are used for whole blocks (that of course should have classes for block tags, usually <p> tags in a book). In other words, removing formatting before conversion saves a lot of time later.

In Font Format - Font: Make sure that the Italic and Small caps are filled, so that these styles will carry over. Font name and size doesn’t matter (since you will not save and embed fonts). Set the rest to Auto or None.

In Font Format - Spacing, set Scale to 100% and the rest Normal. No kerning.

In Paragraph Format, set Indents and Intervals to 0 (zero) or none, set Alignments to Left. Uncheck everything in the Line & Page Breaks tab except Do not hyphenate.

In File - Page Settings, set margins to any size all around, header and footer to 0 (zero). Choose any Page Format (A4 or letter) in the next tab, set orientation to Portrait. Uncheck everything but Supress endnotes in the Layout Tab.

Saving as EPUB

The last step before the fun begins for real: saving the document as an EPUB. Go to File - Save Special - Save as eBook… You can fill in meta data such as Book title and Author if you like, or leave it for later, but one thing is important here: do NOT embed fonts. Select “Don’t save” fonts.

Creating EPUB from a web page

If you want to make an EPUB from a web page out there (only for your own personal use, of course), don’t save the web page using “Save as” or something like that. Right-click and choose to view page source, then select all that text and save it in a text file. Then copy it and paste it into Sigil in code view (chapter by chapter into several XHTML files if needed). That will save you from having to clean up a lot of garbage.

Style Sheets

You have to decide which is your style sheet philosophy: what to decide for others and what to leave be for reader settings. I have drawn exactly that line: none of my style sheets will enforce features that can usually be adjusted using e-reader settings.

However, books that I have designed with my “extended” style sheet *will* suffer if the CSS file is replaced with the “Replacement CSS”. They will still be quite readable, but poems, songs, and letters will look similar to running text when their classes are gone from the CSS.

It is never easy. Create a book that looks very nice, but is still reasonably customizable, or a book with very basic, barely utilitarian design?

Oh well. There are three style sheet versions below: a very basic one, an extended one that contains classes that may come in handy, and an extremely simple one to use as a replacement in books with painfully bad (inadjustable) formatting, or with a GUI-designed book (that will have lots of its styling, such as margins (“indents”) in the XHTML files). Always check any old style sheet for italics before replacing!

Basic style sheet (CV)

```
@page {
margin: 5pt
}
/*
body{
margin-top:1em;
margin-right:1em;
margin-bottom:1em;
margin-left:1em;
font-family:Georgia,serif;
font-size:100%;
line-height:1.5em;
color:black;
background-color:white;
widows:1;
orphans:1;
}
*/

/* = = = = = = = = Standard Elements = = = = = = = = */
h1{
```

```css
text-indent:0;
text-align:center;
margin:1em 0 0.5em 0;
font-size:1.5em;
font-weight:bold;
break-after: avoid;
}
h2{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1.2em;
font-weight:bold;
break-after: avoid;
}
h3{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1em;
font-weight:bold;
break-after: avoid;
}
img{
max-height: 100%;
max-width: 100%;
}

/* uncomment p below if you prefer to have the text justified by CSS */
/*
p{
text-align:justify;
}
*/

/* = = = = = = = Block Style Classes = = = = = = = */
.centered{
text-indent:0;
text-align:center;
}
.right{
text-align:right;
}
.small{
font-size:0.75em;
}

/* = = = = = = = Block Content Classes = = = = = = = */
.firstline{
```

```
text-indent:0;
margin:1em 0 0 0;
}
.text{
text-indent:1.2em;
margin:0;
}
```

Extended style sheet (CV)

```
@page {
margin: 5pt
}
/*
body{
margin-top:1em;
margin-right:1em;
margin-bottom:1em;
margin-left:1em;
font-family:Georgia,serif;
font-size:100%;
line-height:1.5em;
color:black;
background-color:white;
widows:1;
orphans:1;
}
*/

/* = = = = = = = = Standard Elements = = = = = = = = */
h1{
text-indent:0;
text-align:center;
margin:1em 0 0.5em 0;
font-size:1.5em;
font-weight:bold;
break-after: avoid;
}
h2{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1.2em;
font-weight:bold;
break-after: avoid;
}
h3{
text-indent:0;
text-align:center;
```

```
margin:1em 0 0.3em 0;
font-size:1em;
font-weight:bold;
break-after: avoid;
}
img{
max-height: 100%;
max-width: 100%;
}

/* uncomment p below if you prefer to have the text justified by CSS */
/*
p{
text-align:justify;
}
*/

/* = = = = = = = Inline Style Classes = = = = = = = */
.firstletter{
font-size:1.5em;
}
.quotefix{
word-spacing:-0.1em;
}
.ref {
vertical-align:super;
font-size:1.2em;
text-decoration:underline;
}
.small{
font-size:0.8em;
}
/* smallcaps are used when left at the beginning of chapters, where it doesn’t
really matter if they are parsed */
.smallcaps{
font-variant:small-caps;
}
.spellout{
-epub-speak-as: spell-out;
}

/* = = = = = = = Block Style Classes = = = = = = = */
.centered{
text-indent:0;
margin:0;
text-align:center;
}
.left{
text-indent:0;
```

```
text-align:left;
margin:0;
}
.noindent{
margin:0;
text-indent:0;
}
.right{
text-align:right;
margin:0;
}
.xmargin{
margin:0 0 0 2em;
text-indent:0;
}

/* ======== Block Content Classes ======== */
.author{
text-align:center;
font-size:1.2em;
font-weight:bold;
margin:1em 0 0 0;
}
.booklist{
text-indent:0;
font-size:0.8em;
margin:0;
}
.copyright{
text-indent:0;
margin:0;
text-align:center;
font-size:0.6em;
}
.dedication{
text-align:center;
text-indent:0;
font-style:italic;
margin:1em 0 0 0;
}
.disclaimer{
text-indent:0;
font-style:italic;
margin:3em 0 0 0;
}
.firstline{
text-indent:0;
margin:1em 0 0 0;
}
```

```
.footnote{
text-indent:0;
font-size:0.7em;
margin:0.7em 0 0.7em 0;
}
.halftitle{
text-align:center;
text-indent:0;
font-size:1.2em;
margin:3em 0 0 0;
}
.letter{
margin:0 0 0 1.5em;
text-indent:1.2em;
}
/* poem class can usually be used for songs, too */
.poem{
margin:0 0 0 1.5em;
text-indent:0;
}
.publisher{
text-indent:0;
text-align:center;
margin:3em 0 0 0;
}
.scenebreak{
text-indent:0;
margin:0;
text-align:center;
}
.text{
text-indent:1.2em;
margin:0;
}
.title{
font-weight:bold;
font-size:1.4em;
text-align:center;
margin:1em 0 0 0;
}
```

Replacement style sheet (BV & CV)

Assuming that you have an EPUB with <h> tags for headings and <p> tags for paragraphs, you may choose this style sheet (edited to your preferences, or copied as it is) for books “only” edited in GUI, too. That way you can have paragraphs with indents and no margins (blank spaces between all paragraphs). Note that it is important that all scenechanges are in place (by a row of asterisks or similar) before applying this.

```
h1{
text-indent:0;
text-align:center;
margin:1em 0 0.5em 0;
font-size:1.5em;
font-weight:bold;
}
h2{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1.2em;
font-weight:bold;
}
h3{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1em;
font-weight:bold;
}
img{
max-height: 100%;
max-width: 100%;
}

/* uncomment “text-align:justify;” below if you prefer to have the text justified by CSS */
p{
text-indent:1.2em;
margin:0;
/*text-align:justify;*/
}
```

Comments on Styling and Style Sheet Classes

When I started making EPUBs from PDFs, I tried to imitate what the books looked like in paper versions. It was often quite fun trying to come up with working solutions that didn't depend on unreliable things like pseudoclasses and dropletter tags.

As should be clear by now, I changed my mind, and advocate creating books that are easy to customize using e-reader features such as choosing fonts, font size and font weight, line-height, and margins. My style sheet settings are deliberately simple in order to make sure that they work for the majority of e-readers.

Don't style an EPUB like a web page

When I had just started with e-books and discovered that EPUBs were really "just" archives of XHTML files, I was delighted. I had worked with web pages for several years, and also had a web developer education. I knew that "I know this!".

Wrong. Very wrong. Bringing the good practices of web pages to EPUBs is actually a mistake in many cases. You may know all about HTML and CSS, which certainly helps, but you still need to realise that EPUBs are not web pages.

I had been taught that:

- A text looks best if it is justified (same margins to the left and to the right) - WRONG!

This may be true for web pages to be read on large computer screens, but it creates problems on EPUBs, especially for people with dyslexia. When the text is justified, the spaces between words are different in order to make the side margins even. Increase the font size a bit, and this becomes very obvious. Justification may make the text much more difficult to read.

So, the best EPUB practice is to not justify the text. It will then usually default to left-aligned.

Most (all?) e-readers allow the reader to justify the text if they want to. If not, people will have to learn how to open an EPUB in Sigil (or other editing software) and make the text justified. In my style sheets, it is merely a matter of uncommenting the p setting.

- Non-serif fonts are easier to read than serif fonts - WRONG!

This may, again, be true for web pages designed to be read on large screens, but is not true for novels. There is a reason why most paper books are set in serif fonts.

There is good reason for not specifying e-book fonts at all. Leave it for the reader to decide.

- Increased line height makes texts easier to read - WRONG!

Again, it may be true for web pages, but this only wastes space for EPUBs. In addition, many e-readers allow you to increase the line height yourself, if you should prefer that. Leave it be, don't force an increased line height on the reader.

- The default blank space between paragraphs makes it easier to read long texts - WRONG!

This may (again!) be true for large screens with walls of text, but not for smallish e-readers. EPUB novels are better off looking like paper books in this respect. (Non-fiction books like this one, on the other hand, may benefit from keeping the blank spaces between paragraphs.)

So: remove the default margin (blank space) between paragraphs and add an indent for new paragraphs instead (or else the EPUB text will be too compact).

Don't style an EPUB like a paper book

Some paper books have very decorative chapter headings. I used to mimic those, and indeed often had fun trying to come up with ways to achieve this with simple code that should work in all e-readers. Lots of "special" coding, such as dropcaps and spacing will not work in many e-readers, so I resorted to left-floats and specific paddings and margins and line heights... And then I discovered that even the simple work-arounds tended to run into problems in different e-readers.

The truth is that it really does not make the EPUB better if the headings look the very same as they did in the paper books. It can indeed be very very difficult to make special headings (and firstlines) look like the paper book without specifying fonts, font-sizes, and line heights, etc.

In other words: in order to make the start of chapters look like the paper book did, you may need to make the book practically unusable for readers who need to be able to select fonts, font sizes and line heights for themselves. It really is not worth it. Just use the ordinary h1, h2, h3 (etc.) without over-complicating them.

Some paper books are printed with very large side margins. I have seen this mimicked in quite a few commercial EPUBs. Madness, in my opinion. What looks pretty on paper is just a waste of space in an EPUB - and it will look way worse to the reader that needs to make use of the magic of e-books and increase the font size. This tends to increase the margin size, too, leaving very little room for the text.

And, of course, most paper books have justified text. Do not mimic it!

On the other hand, you may actually want to add some styling that isn't there in the paper book. I am specifically thinking of letters, songs, poems, and other paragraphs that should be read in an "alternate voice". The margins often used in paper books may well be applied to the EPUB, too, but in addition you may want to add <i> tags.

In all. Use your common sense and try to think of different perspectives...

Classes and styles in style sheets and XHTML files: a brief explanation

Apart from the standard elements, such as "h2", "img", and "p", I have added a number of classes to the style sheets. You see that they are classes by the dot before their names, such as .firstline and .text. When a style sheet entry is defined as a class, it can be used many times in the same XHTML file (unlike an ID, that has to be unique).

A style sheet entry has no effect if there is nothing in the XHTML files that corresponds to the entry. If there are no <h2> or <h3> tags in the XHTML files, the h2 and h3 settings in the style sheet will have nothing to affect.

You make a style sheet class entry like .firstline take effect by adding it inside a tag in a XHTML file. By changing <p> to <p class="firstline">, you add the styling and formatting described for ".firstline" in the style sheet to that paragraph, which will make it look different from paragraphs using <p class="text"> or <p class="copyright">, etc.

Inline classes are used inside paragraphs, notably in <span> tags, such as <span class="small">some words</span>, and affect the words between the opening and closing tags only.

You can add styles in three places: in a separate style sheet (a CSS file), at the start of a XHTML file, or directly preceding the item that the style will be applied to. There is a hierachy here: if there are conflicting styles at different places, the one closer to the item will “win”.

It is preferable to not have any conflicts at all, of course. This is why I have previously done all the styling in the CSS file. However, I have realised that such practice makes for poor accessibility (especially with EPUB3), so styles that may have semantic meaning will have to be added as tags to the XHTML files instead. Italics and bold text that is merely some kind of decoration for titles, dedications, etc. can stay in the CSS file.

How to name classes

Software for converting to EPUB will sometimes (often!) create a style sheet, CSS, as well, and fill it with various classes that are then used in the XHTML pages for styling the book. Usually they have names such as p1, p2, t1, block10, calibre1, calibre2, etc.

You can check the XHTML files in order to see which classes are used for what (hopefully consistently so), but it really is much easier to edit a book with intelligible names for the classes. So, when you have a choice, I suggest you give the classes descriptive names instead.

Comment out

You can make comments in CSS (as well as in HTML and most other codes). A commented style sheet line (or text block) starts with /* and ends with */. A commented line may contain settings, but they have will have no effect, unless/until the line is uncommented. See, for example, the “body” entry in my style sheets: I leave it commented out, but if somebody who gets one of my books wants to specify margins, background colour, etc. for the whole book, they can uncomment that part and edit the settings.

Uncomment

So, a commented line (or text block) in CSS starts with /* and ends with */. Remove those, and the line (or text block) is uncommented and goes live, making the settings affect the HTML element it is applied to.

@page

I set a small margin in fixed size pt since a relative unit (such as em or percent) would make the margins very large if the font size were much increased by the reader. I set 5 pt all around so that the text will be readable even if the physical e-reader has bevels that may otherwise shade text close to them. I set these margins in @page because that way they will apply to all "pages" in a physical e-reader. Top and bottom margins for <body> in CSS only apply to the top and bottom of the XHTML file. There can be many "book pages" in one single XHTML file. Also, using @page rather than margins for elements allows for choosing the width of (side) margins in the e-reader.

.author, .title., and .publisher

...are used on the title page. Manual replacement needed, of course, but it's just the three paragraphs.

body

The body element is commented out in order to ensure maximum freedom for the reader to specify margin, font, line-height, colours, etc. It's there as a service to those who might want to edit the settings. The style sheet does not affect epub:types like "bodymatter" that is added to the XHTML pages, no matter if it is commented out or not.

.booklist

Used when there's a page with "also by" or similar. Replace <p> with <p class="booklist"> on current page (after splitting).

.copyright

Replace <p> with <p class="copyright"> on current page. This assumes you have split pages, of course, so that copyright is on its own page. Add blank lines as in the paper original; not necessary, but it looks nicer.

.img

```
img{
max-height: 100%;
max-width: 100%;
}
```

...will make sure that images don't overflow, and so will not go too large for the reader.

.firstletter

Simply making the first letter in the first line beginning a chapter larger than normal text; in the extended style sheet it's 1.5 em.

.firstline

…is not really necessary, but IMHO it is nice to use it for the start of chapters. The rest of the paragraphs are indented, but this firstline is not.

.footnote

The text of the footnote, at the bottom of the XHTML file.

p

The style sheets contain a paragraph setting for justified text that you only need to uncomment if you want justified text and your e-reader doesn't support making it so. The setting is commented out for accessibility reasons: justified text may be hard to read if you need very large fonts, and the irregular spaces between words may also make the text harder to read for people with dyslexia. Many e-readers allow you to set text alignment according to your preference - but this will only work if "text-align" isn't specified in the style sheet (or by style in the XHTML file).

.quotefix

This is used for single quotes directly after double quotes, or vice versa. It is very far from necessary; I use it because it's neat and easy to apply with a normal search; it "ties" single and double quotes together with a non-breakable space while reducing the distance between them a little, so that a spoken line starting or ending with somebody quoting somebody else looks good. Happens a lot in mysteries! There are instructions for applying it in the Corrections chapter.

N.B. this is a simple formatting work-around. The most elegant solution would be not to use spaces between them, but increase the spacing so that they don't run into each other. Unfortunately it seems that some e-readers will not parse spacing settings, so work-around is it. If the "quotefix" isn't parsed by an e-reader, the non-breaking space will still prevent a single quote from ending up at a new line.

.ref

Used for the link (usually a digit) to a footnote. Underlined superscript, somewhat larger than the normal font so that it’s easier to click.

.smallcaps

The class is used when leaving smallcaps in, for example at the start of chapters.

The small-caps font variation is unreliable: not all e-readers will parse it. I usually leave them be if they are in the start of chapters; it does no harm if they aren’t parsed. I change to upper case with a small span class for signs and similar, that needs to be in “caps” no matter if the reader can parse smallcaps or not. Note that there is a button (“AB”) in Sigil for quickly changing all letters in a selection to upper case, so it is quick and easy work, then add/change the span class to “small”.

.text

...is the class I use for everything that could be left plain <p>. In most cases it is not necessary, but there are some readers that won’t parse CSS settings for plain <p>, and so they will be stuck with blank spaces between paragraphs no matter the margin:0 setting in the style sheet. To use, just replace <p> with <p class="text"> on all XHTML files when all other styling is done. Then the text paragraphs will be indented, with no margins (blank spaces) between them.

Semantic Elements and Tags

Invisible tags

Basically, many new semantic tags in HTML5 will be left invisible to the human reader. You *can* style them and so make them visible, but their main purpose is to provide information and structure to machines, such as screen readers. Making good use of these tags means making the reading experience better for visually impaired readers.

EPUB types are attributes that, as suggested by the name, tell screen readers what type of content there is in the XHTML file (when applied to the <body>) or a part <section> of the file. Since I advocate one file for each type of content (copyright page, dedication, chapter in book, etc.), it will usually be easy to decide if it is a question of adding the EPUB type "frontmatter", "bodymatter" (the payload, as it were), or "backmatter" to the <body> element.

The <**section**>, with EPUB types such as "chapter" and "dedication" provides more detailed information about the content to the screen reader. You may also occasionally use <**figure**> and <**aside**> as containers for images and footnotes, respectively.

Appendix 2 lists EPUB types and where to add them in approximate book order.

Acronyms (abbreviations): the <abbr> element

It is a little tricky to decide what to do with some acronyms when considering how a screen reader will interpret the information. It seems that there is no consistent interpretation of the tag. So, using the tag for XHTML and similar acronyms that should be spelled out as a line of letters seems to be touch-and-go. Then, what about acronyms/abbreviations that are pronounced as words? Also unclear. "EPUB" shouldn't be pronounced as "ee-pee-you-bee" but as a place where you could get an electronic glass of beer, and "MOBI" similar to the first name of a famous whale.

Web searching has informed me that that you can create a class for spelling out acronyms in CSS3, and then apply it to acronyms like XHTML to have them spelled out letter by letter, which suggests that it is indeed not certain that plain <abbr> will lead to such spelling out. However, my guess is that it will vary, so this is what I will do:

No tag for acronyms/abbreviations that should be pronounced as words, such as EPUB and MOBI.

<abbr class="spellout">XHTML</abbr> for acronyms that need to be pronounced letter-by-letter, along with a class like this in the style sheet:

```
.spellout{
-epub-speak-as: spell-out
}
```

If an explanation is needed, I will add it in the text the first time the acronym appears, like this: “EPUB” is short for “electronic publication” (which is why you will occasionally see it more correctly written as “ePub”). To be sure, you will seldom need to address this problem in your average scan of a paper book, as any needed explanation should already be in the text.

> *There is of course another way. Used on its own, with or without the spellout class, <abbr> just tells the machine that this is an acronym. If you add a title, such as <abbr title="Table of Contents">ToC</abbr>, hovering over the visible word (ToC) will show the full expression as a tooltip — which can be useful for web pages, but pointless in EPUBs.*

Headings (<h1> etc.)

Make sure that the hierarchy of your headings is correct. There should, at most, be only one <h1> heading on a XHTML page (file) - I would argue that you should save the <h1> for books that are divided into parts, and then use <h2> headings for chapters, and continue with <h3> headings for subchapters, etc. That way, a screen reader will get correct hierachy information, as will any visual reader (assuming logical design of the headings).

Tables

Make sure that you use tags to properly show the structure of any table you may need to insert. They are not common in novels, of course, but I have had to take recourse to tables once in a while when detectives in old mystery novels insist on creating lists with columns showing where the suspects were when the crime was committed, what their motives might be, etc. For headers (such as Means and Motive), use <**th**>, and make use of cells (<**td**>) inside rows (<**tr**>) rather than using blank spaces or similar to separate different kinds of information.

The tags <i>, <b>, <em>, <strong>, plus "italic" and "bold" in CSS

In EPUB2 (HTML4), <i> and <b> show as italic and bold text, respectively, with no semantic meaning, while <em> adds emphasis and <strong> adds more emphasis.

In EPUB3 (HTML5), <em> still adds emphasis, but <strong> now adds importance/urgency rather than stronger emphasis. For example, you would tag a question "Do you <em>really</em> mean that?" in order to suggest doubt by emphasising the word "really". If you wanted to have more emphasis, you would use tag nesting (yes, really!). Like this: "Do you <em><em>really</em></em> mean that?"

You would tag a warning or other urgent message like this: <strong>Here there be dragons</strong>. And, yes, you can nest <strong> tags, too, for more important and/or urgent information.

The semantic tags, <em> and <strong>, may be shown in text as italic and bold, respectively, and so look no different to <i> (span class or style "italic") and <b> (span class or style "bold"), but they do not have to follow that convention. They *could*, for example, be colour-coded.

So, what about <i> and <b>? I have read books and searched the web. The official sources say that <i> indicates an alternate voice or mood, and can also be used for names in HTML5 (and so, presumably in EPUB3), but it is not really clear that this translates to semantic information that assistive technologies such as screen readers can make use of. Other sources claim that <i> still adds no semantic information, and so merely makes text italic. I have concluded that <i> adds semantic information if you can see and interpret it, but that it will not be of any use to a screen reader.

Similarly, <b> is sometimes recommended for "keywords and other semantic uses" while not conveying voice or mood change (like <i>). I rather suspect that this is another case of a tag that is still not adding semantic information to a screen reader. It just adds bold text that can be interpreted by a reader that sees it.

So. In EPUB3, you will use <em> and <strong> as indicated above in the XHTML files. The case for <i> and <b> is unclear, but it will do no harm, and may be useful, to apply them in XHTML files rather than style sheet classes for letters, songs, poems, and quoted thoughts, etc. Just don't rely on them to convey semantic information to screen readers.

You can still add merely decorative and presentational italics and font-weights in style sheets, for example making dedications show in italics and the book title in bold. A screen reader will not pay attention, but the visual reader may well appreciate the styling.

The <i> tag for a foreign language

Books in one language will occasionally have words or paragraphs in another language. This can be indicated to a screen reader by using the “lang” attribute. Like this, for a French word in this otherwise English book: <i xml:lang="fr">enchanté</i>. It will show as *enchanté* — only the screen reader will know that it is indeed in French. Other foreign languages? There are plenty of lists of language codes on the web. Search and you will find.

The <cite> tag for book titles and other publications

The previous (HTML4/EPUB2) standard was to use italics (<i> or a style/class) for book titles, and some HTML5/EPUB3 “experts” still say so. However, in HTML5/EPUB3, you should use the <cite> tag instead. It will tell a screen reader that a book title is indeed a book title. Like this: <cite>E-Books That Cut It, 2nd ed.</cite>. Use it for other publications such as papers and magazines, too.

Do you really have to decide between <i> and <em> (and <cite>)?

All you see on the book page is that some parts are in italics, and it struck me that that sight beautifully illustrates the situation of the screen reader. There is no way to tell if you are seeing decorations, or emphasis, or letters, without some semantic information.

This is why I think it should be worthwhile to go through the italics of the scanned text, and correct the tags. It is boring, yes, but it will make the finished book that much more usable to people who are not visual readers.

In most novels, most of the italic text should be <em>-tagged, since it is very common to use italics for emphasis. There will, however, be excerpts from letters, etc., so you cannot simply replace all italic span classes or <i> tags with <em> and leave it like that. After replacing with <em>, run a search and change to <i> (or <cite> for book titles) as appropriate. And add language attributes as needed.

ARIA (Accessible Rich Internet Application)

In addition to making the best possible use of the semantic opportunities in HTML5, you could learn to add ARIA roles, which enhance accessibility when using screen readers and others assistive technologies.

However, ARIA is quite difficult to learn, and will be the opposite to helpful if applied incorrectly. I will stay away for now, but might return in the future.

Working with Sigil GUI (Book View or PageEdit)

Introduction

You don’t *have to* work in Code View. You have more control over the final product if you apply your knowledge of HTML and CSS in Code View, but using the GUI will go a long way. Especially compared to just converting with ABBY (or Calibre!) and leaving it like that.

N.B. that it will be difficult/impossible to add most of the semantic elements and tags discussed earlier when working in Book View or PageEdit. You can still convert to EPUB3 and validate it, but not add that extra layer of accessibility.

One problem with the GUI is that you can’t, for example, apply the zero margin and the indent you might like to use for most of your paragraphs (<p>) in a novel. There will be spaces (like empty lines) between all your paragraphs, and there will be no text-indent in paragraphs.

You also can’t change all the headings in a way that suits you. You can certainly click and add italics, or centering, etc., but all you get is tags for the individual clicked instance. You have to click all your h2 headings, one after the other, if you want them to look a certain way.

In short: you can’t specify styles when using the GUI. That doesn’t have to be a problem. Just go with the defaults, and you’ll be fine…

Or, provided that your EPUB contains ordinary <h> and <p> tags, use the “Replacement CSS” from the style sheets chapter.

However, using Sigil’s GUI with a style sheet with your specific formatting decisions will sometimes lead to unwanted changes in the design. Sigil reads the style sheet settings and may apply them as styles in the text when you remove lines or move parts of the text. Paragraphs may get styles that then cannot be edited without going into Code View in order to see just what happened.

You may end up with a heading going larger than the other ones of the same level because Sigil added a 1.2em span style to the existing 1.2em font-size in the style sheet, or a heading getting a paragraph class when moved up by deleting previous lines, or indeed a heading changed into a paragraph with the heading attributes set in the shape of span styles; that (previous) heading will not be included when you generate a new ToC.

> *You can add styles at three locations: inline, i.e. in the text of the XHTML file (along with the HTML tags), embedded at the top of the XHTML file, and in an external style sheet (a CSS file) that the XHTML file links to. Styles in the text of the XHTML file override styles at the top of the page as well as styles in a style sheet/CSS file. (Styles at the top of the page override styles in a style sheet/CSS, but not those in the text.) So: styles in the text of the XHTML file can mess up what you want to accomplish with a style sheet.*

This is why the workflows for Code View and Book View are different when it comes to adding a style sheet. **When working in Book View, only add the style sheet when all corrections are done**, assuming that you don't want blank spaces (margins) between your paragraphs, and also want a text-indent for those same paragraphs (which makes novels nicer to read). You can correct misspelled words without getting unexpected style changes, so there is no problem with adding the style sheet before proofreading.

Of course, you can temporarily add the style sheet before that, in order to check what the book will look like. But do remove it before you continue editing.

Below are good practices to keep in mind all the time. Notably, saving searches and clips saves a lot of work, while saving a copy at the right time might save you from disaster.

Save a copy

There is an option to save a copy in the File menu. Use it before making thorough changes, such as using Replace All when correcting problems. Such a copy may prove very valuable if the result turns out to be undesirable, but you noticed too late.

Searching

Go to Search → "Find & replace…" in order to open the searchbox (at the bottom of the window).

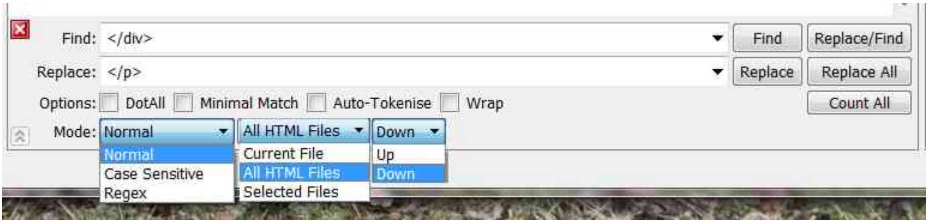
When you want to search the CSS or apply replacements to one page only, make sure that you have selected **Current File**. When you want to search the whole book, select **All HTML Files**.

When you search with **Regex**, it simply will not work unless you have set the search mode to Regex. Normal searches will usually work in Regex, too, unless your search terms happen to include any of the "special characters” that have specific meanings in Regex. Best to switch to Normal mode for normal searches!

Normal mode (or **Case Sensitive**) is still “only” for the text as you see it in Code View.

If you stay with the default **Down** direction of searches, make sure that you start your search at the top of the page when searching Current File, and at the very start of the book when you search All HTML Files.

This is an illustrative fake image: you can’t have all the selections showing at once without faking it:



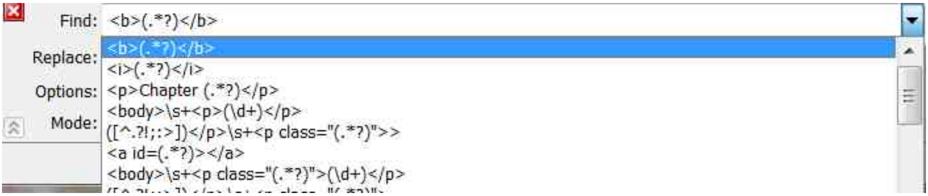
Find: </div>

Replace: </p>

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Normal | All HTML Files | Down

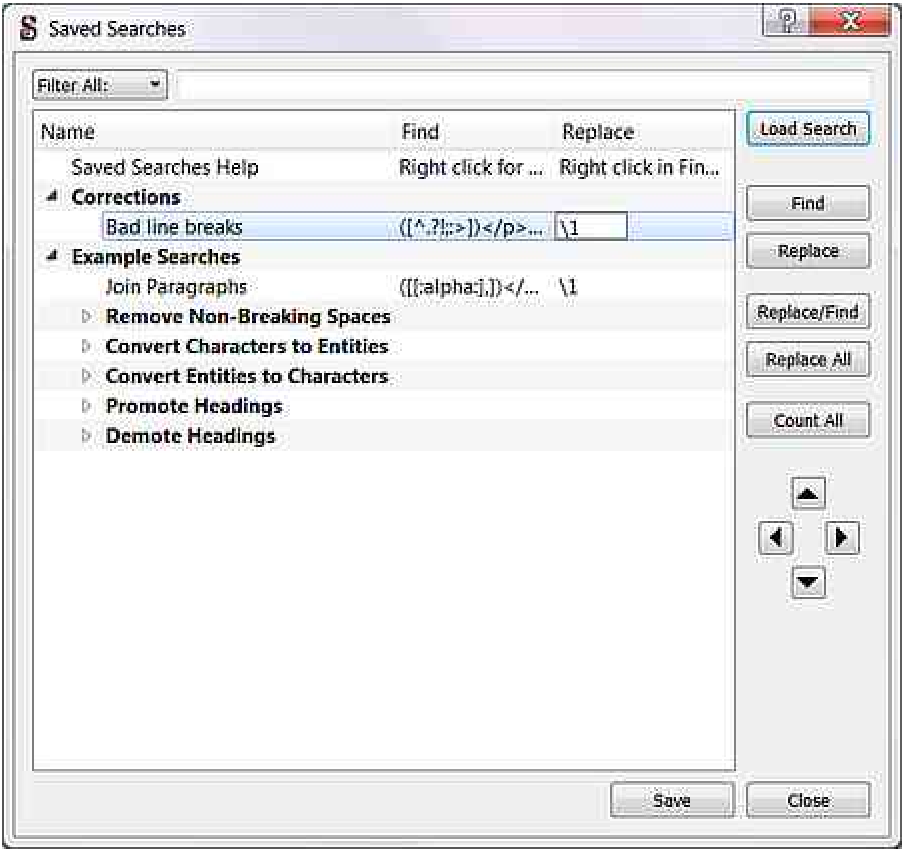
The latest searches are saved in reverse order of appearance (latest saved searches first). This is of course very useful for searches that you repeat many times: just click the little down arrow to the right of the text field (either “Find” or “Replace”) and select the search term you want to use.



Find: <b>(.*?)</b>

```
<b>(.*?)</b>
<i>(.*?)</i>
<p>Chapter (.*?)</p>
<body>\s+<p>(\d+)</p>
([^.?!;:>])</p>\s+<p class="(.*?)">>
<a id=(.*?)></a>
<body>\s+<p class="(.*?)">(\d+)</p>
```

You can also add searches by going to Tools → “Saved Searches…”. There are a number of pre-defined searches there already. You can add and delete Groups and Entries by right-clicking in the window. Here I have added a “Corrections” group, and a search for bad line breaks:



Name	Find	Replace
Saved Searches Help	Right click for ...	Right click in Fin...
◢ **Corrections**		
Bad line breaks	([^.?!;:>])</p>...	\1
◢ **Example Searches**		
Join Paragraphs	([[:alpha:],])</...	\1
▷ **Remove Non-Breaking Spaces**		
▷ **Convert Characters to Entities**		
▷ **Convert Entities to Characters**		
▷ **Promote Headings**		
▷ **Demote Headings**		

The searches are stored in a file called “searches.ini” or “sigil_searches.ini”, that you can copy and paste into the Sigil directory for use on another computer; the location varies with OSes and versions, but the ini file seems to work everywhere there is a Sigil.

Saving useful text to paste

There is a **Clip Editor** under “Tools” where you can store often-used phrases (which is perhaps most useful when working in Code View, but deserves a mention here, too). Right-click to add Groups and/or Entries, the same way as in Saved Searches. You can access your clips (and useful example ones) with a right-click in the text you are working on.

The clips are stored in a file called “clips.ini” or “sigil_clips.ini”, that you can copy and paste into the Sigil directory for use on another computer; the location varies with OSes and versions, but the ini file seems to work everywhere there is a Sigil.

Preparation

This workflow will be described with Book View in mind. It is basically the same if you use the free-standing PageEdit. Note that you have to open your EPUB with Sigil even if you use PageEdit for GUI editing, and that you can only work on one XHTML file at the time in PageEdit.

What follows are detailed instructions for what to do if you work according to the previously outlined workflow. The next three chapters "only" deal with structure and design, though. The last parts of the workflow, corrections and finishing touches, have got their own chapters.

Copy and edit cover and images

With the PDF open in ABBYY (or other PDF software), copy the cover and save the image in Paint (or other program that allows you to paste and then save in bmp, tif, or possibly as png). Open it and fix as necessary in Photoshop, GIMP, or other similar software. Crop the cover if needed, change brightness/contrast if needed, do other fixes as needed, then save in the original format (such as png, bmp or tif). If the cover is in too poor shape, leave it as it is or go hunting the webs for a better image. N.B. if you need to replace colours, for example in a map that has a brown or grey background that should be white, I recommend leaving it till the IrfanView stage (below); in my experience, it works better than Photoshop.

Sometimes the cover image of a PDF downloaded from Gutenberg or similar repository will be impossible to copy properly (maybe because of the PDF created in A/archive format). The simplest work-around that I have found is to open the PDF in Photoshop, and then work with the cover image from there. It should be possible to open it in GIMP as well.

Also copy the title page if it looks nice enough to insert as a fullpage image instead of making a plain text title page. Edit in Photoshop and then replace background colour as needed using IrfanView.

In addition, copy other images, if there are any, such as maps or small illustrations, and edit as detailed above and below. The size of images that are smaller than fullpage will of course depend on the context; it's a bit of trial and error, so make sure that you keep the original.

Once edited in Photoshop (or GIMP, etc.), open the image(s) in IrfanView. IrfanView is much better for downscaling with decent quality than Photoshop or GIMP.

If you need to replace colours, open Image - Replace Color and click on the color in the image that you need to replace. It will be automatically added in the source colour field. Then select the new colour. You may need to toy with the tolerance value a bit in order to have the best possible result.

If applicable, then click Image - Convert to Grayscale.

Finally, click Image-Resize/Resample and resize to something suitable – I usually downsize cover and fullpage images to a height about 750 pixels. Make sure that the option to preserve aspect ratio is checked.

Save in jpg if the image is photolike, with shades and nuances, in gif if there are clearly contrasted full-colour fields. The quality can be low, about 30-60 of 100 or so; this takes the file size down.

Open your EPUB in Sigil

Note that you can change between Book View and Code View by clicking the open book and < > buttons above if using an old version of Sigil, or else open a XHTML file in Page Edit by clicking the square icon that appears to be crossed by a pen.

Remove the text in any existing style sheet

As mentioned above, style sheet settings can be applied in unwanted ways when editing in Book View. So, open the Styles folder and double-click the CSS file, copy all the text there to an ordinary text file and save it, then remove all text from the style sheet in the book. If you do want to apply styles by style sheet, such as having text-indent for paragraphs in a novel, add the style sheet when all the editing and all the corrections are finished. The step is included in the last chapter ("The Finishing Touches").

However, if you are not working with an EPUB created by exporting from ABBYY, but one created some other way (or downloaded as EPUB), you have to check the style sheet first. **If the italics in the book depend on that style sheet, removing it will cause a small disaster.**

In such cases, you need to search the style sheet for "italic" settings, and leave those classes in the style sheet, now and when you paste your own one when the book is finished. (Changing them to <em> or <i> would require working in Code View.)

Search and replace all &nbsp; and &#160; with ordinary spaces

ABBYY makes lots of them. IMHO it’s OK to use &#160; as needed while designing, but these are superfluous, so get rid of them first.

If the search box is not open, go to the Search menu and click “Find and replace…”. Use normal mode and search all XHTML files. Assuming that the default direction “Down” is used, place your cursor at the very first line of the book before you start.

Find:	**Replace with:**
&nbsp; or &#160;	a blank space — **it is very important not to forget this**, since no space here means words running together…

Remove HTML ToC in text file and generate ToC

Look for a table of contents with links that is the result of a ToC in the paper book and remove it, if you find one. *(Yeah, yeah, I know. It would be hard to remove a nonexisting ToC, but oh well. ABBYY will not always insert ToCs, so the question may arise.*) Search for <ul> if you can’t spot it right away.

Generate a ToC instead, either by clicking the icon somewhat top right, or by going to Tools → “Table of Contents” → “Generate Table of Contents”. Do this right away (and then again as needed, and again when the book is finished); often ABBYY will have added <h> tags that don’t show in the ToC when the book is first opened after conversion.

If you are creating a EPUB3, update the toc.ncx for EPUB2 compatibility by clicking Tools - Epub3 Tools - Generate NCX from Nav.

The nav.xhtml file is a different matter. It should be here, do not remove it!

Create main structure

Zoom out the PDF for easy scrolling and scroll through the PDF and EPUB side-by-side as it were.

Split XHTML files

If an introductory page has not got its own XHTML file (as it ideally should), you can easily split the file(s): place the cursor at the right place, then hit Ctrl + Enter. These may be title page, copyright page, dedication, epigraph, and others. One file each!

Create a file for the text title page even if you plan to add a large title page image later on. This way, you will have a page in which to insert that image.

(You can also place markers and then perform numerous splits in one go. Place the cursor at the right place, then go to Insert → “Split Marker”. When all the markers are in place, go to Edit → “Split At Markers”.)

If there are extra pages at the end of the book, such as booklists and back cover text, split them into their own files, too.

Don’t worry if the names of the new XHTML files are not pretty. The names don’t matter — though I personally usually rename the special pages for easy overview.

Index

If there is an index and the OCR has left it too messy, I simply delete it. (N.B. Nuance often performs better than ABBYY on indexes, lists and tables, so if you have access to it it might be worth OCRing such pages with it, if ABBYY has made a mess.) If it looks OK or needs very little editing, I leave it, but make no effort to make it “live” by linking. I leave it in order to show what the author found worth indexing.

Split XHTML files as needed, one for each chapter

Split larger files resulting from the conversion. Search in the EPUB for an unusual word near the chapter beginning in order to quickly find it.

Split the same way as you did for the introductory “special” pages: place the cursor at the right place, then hit Ctrl + Enter. Don’t worry if the names of the new XHTML files are not pretty. The names don’t matter.

An EPUB will work even if you don’t bother to have one XHTML file per chapter (and one XHTML file per special page), but apart from being neat, it also makes conversions to MOBI or AZW better, since they tend to “look” at XHTML files.

> *Optional (if you are orderly, like me:) if there are orphaned XHTML files due to splitting at chapters, copy and paste the text into the correct chapter XHTML file, then delete the superfluous files. One XHTML file for each chapter.*

Add cover

Go to Tools → “Add cover...” and browse to your saved (jpg or gif) image. Sigil will create a new cover.xhtml file, placing it first in the list of XHTML files. N.B. if your cover isn’t already called “cover”, right-click the image in the Image folder(in the Book Browser to the left) and rename it. It will not show properly in Calibre (and some other software) if it is not called “cover.jpg” (or, cover.png, or cover.gif, etc.)

Add XHTML files

Sometimes (rarely), you will need to add XHTML files in Sigil, for example in order to insert a fullpage map into the middle of the book.

Right-click the XHTML file above the desired place of the new page, and choose “Add blank HTML file”. It will be called “section-something” and there’s no need to rename it. It is the order of files, as shown in the left-hand book browser, that is important.

The new file will have a very basic head section. Notably, the link to the CSS file will not be copied, and the result is of course that your CSS (if you add a style sheet) will have no effect there. So, go to an “old” page in Code View and copy the top section of the XHTML file, from </head> and upwards. Typically it will look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns:epub="http://www.idpf.org/2007/ops">
<head>
  <title>Pride and Such</title>
  <link href="../Styles/main.css" rel="stylesheet" type="text/css"/>
</head>
```

Paste it into the new page (in Code View), replacing the old <head> to </head> section.

See “Fullpage images” in the “Main book design” section for how to insert such images.

More images

If you spot any **images** that you haven’t already copied, copy them from the PDF, fix and rescale them, so that you can insert them in the EPUB where they should be – and insert any already copied and fixed images you saved earlier.

You insert images by going to Insert → “File…” and then clicking the “Other files…” button. Browse to where you have saved the image to be added. If you have downscaled and saved as jpg or gif as instructed, make sure that you don’t accidentally add the original, needlessly large, one instead.

Fullpage images (copied title page, large map, etc.)

In order to make fullpage images cover a full page, while keeping correct ratio and not overflow (turn larger than the e-reader can show), you need to go into Code View and apply some changes to a (new) blank XHTML file.

Use Insert - File... first, inserting the image in its own otherwise blank XHTML file, so that the image is added to the book, then edit the page. If you don’t know the size of the image, open the Images folder in the Book Browser pane to the left and double-click the image file name; the information is there.

A cover added using the Add Cover tool is marked up like this; I use it for other images that need to be full-page, too, just changing the measurements (two places each, width and height in the “viewBox”, then height and width for the image) and of course the file name. The bold text shows what will need to be changed.

```
<div style=“text-align: center; padding: 0pt; margin: 0pt;“>
<svg xmlns=“http://www.w3.org/2000/svg” height=“100%”
preserveAspectRatio=“xMidYMid meet” version=“1.1” viewBox=“0 0 472 750“
width=“100%” xmlns:xlink=“http://www.w3.org/1999/xlink”>
<image height=”750“ width=”472“ xlink:href=”../Images/cover.jpg“ /></
svg>
</div>
```

Design details

The workflow below is approximate, as it has to be: you will find different items to design in different books, and restructuring the order of the workflow will sometimes make it all easier...

Special pages design

The first pages of the book may be titlepage, halftitle (just the title of the book, while “titlepage” usually has author name and publisher name, too), copyright page, maybe credits and a dedication. Apply bold and italic styles as needed (compare to the PDF) by selecting text and clicking the buttons above. If there is a source for an epigraph or a quote, it usually looks best if you align that paragraph to the right.

Continue with similar edits to the special pages at the end of the book (if any), such as book lists or back cover text.

Make the chapter headings h2 or h3

Select text for a heading, then click one of the h1-h6 buttons. Perfect markup, much easier than adding <h2> (etc.) tags manually.

Generate ToC again

Do this when all the chapters have got their headings (and their own XHTML files), and then check that all the chapters are there. You can add or remove items in the ToC, and also edit the entries by going to Tools → “Table of Contents” - “Edit Table of Contents…”. This is useful when the chapter headings are so long that the ToC becomes hard to read.

Remember to generate NCX from Nav if you are creating an EPUB3.

Special styles such as letters and poems

If and when you spot any deviant items, such as **telegrams, letters, poems, and songs**, style them so that they stand out from the running text, similar to but not necessarily the exact same way as they do in the PDF. I would suggest an increased margin. Select the start of a paragraph and clicking the button with an arrow to the right above.

For accessibility reasons, sections that are in an "alternative voice" can be styled as italic by using the <i> tag, which is applied if you select text and click the italic *A* button above.

For visual difference, you can also add blank lines above and below the telegram or letter (etc.).

Smallcaps

If there are SMALLCAPS, just leave them be if they are at the start of chapters. They will not be parsed by all e-readers, but that will do no harm there. If there are smallcaps in the running text, for signs or telegrams or newspaper headlines, etc., select the words that should be in caps and click the “AB” button somewhat to the right above. That will turn all the letters into UPPER CASE.

Add scenechange lines

…whenever you see a scenechange indicated in the PDF. **This is important**: abrupt changes with no indication makes for a poor reading experience.. Indicate scenechanges by adding a new line with some visual clue such as * * *. Center the asterisks by selecting them and then clicking the centering button above.

> *A scenechange (aka scenebreak) is when, for example, a new paragraph takes place in Paris instead of London. Most books have a blank line or some little image, or one or more asterisks, between such paragraphs, so that the reader will be made aware that there is a “jump” in the text.*

Footnotes

Some novels contain footnotes, though usually not as many as there can be in text books.

The easiest way of dealing with them is to place an asterisk where the reference is, then select it and click the A^2 button so that it becomes superscripted. Like this: *. Then place the footnote text it refers to in its own paragraph right below the one with the reference, adding “* [FOOTNOTE]” before the footnote text. No linking involved.

* [FOOTNOTE] So, here is the text to read when you have spotted the * in the paragraph above.

If you prefer to place the footnote texts at the chapter (XHTML file) end instead, with linking back and forth, here’s how to do it:

Place the footnote text at the bottom of the XHTML file. Add “1” or “*” or whatever you will use as a reference in the text at the start of the paragraph.

Select the “1” or “*” and click the anchor button. You will be asked to “insert ID”. You need to give the anchor you are about to create a unique name, as these anchors are used as targets when linking back and forth. I usually call the anchors at the bottom of the page “back1”, “back2”, etc., since they will later be used to link back to where the footnote reference is in the running text.

Next, type the reference (“1” or “*”) in the text and click the chain button next to the anchor. This will provide you with lots of targets to choose between, because all the headings in the ToC will show up. Scroll to your newly made anchor (“back1” in my case) or use the search feature in order to find it. Select it and then click OK. The reference* now links to the footnote text.

But, you want to be able to go back to the text after reading the footnote. So: select the reference in the text again, but this time click the anchor button. As before, you need to choose a unique name; I usually use “note1”, “note2”, etc.

With that ID in place, scroll down to the bottom of the page and select the “1” or “*” you used as the footnote text anchor. Click the chain button and scroll through available targets; select the one you created for the reference (“note1” in my case). You may want to add a line (just press the __ key) above the footnote text, making it clear that it is not part of the running text.

Unexpected and unwanted markup when editing in Book View

I have noticed that Sigil will sometimes create a <div> instead of a <p> when you add a new line by pressing Enter in Book View. These <div>s will of course not be affected by the style settings in your CSS, so you want to change them to <p>. If you spot it, and are comfortable with the occasional visit to Code View, go there and simply select the <div> paragraph and click the “P” button somewhat to the left above. That should change it all from <div> to <p>. You can also run a Normal search for <div>, and change them to <p>, and then </div> to </p> the same way — if you know that you haven’t used <div> in the book for some legitimate purpose. N.B you need to run both searches before saving, or Sigil will complain about errors and offer to fix them in unforeseeable ways.

Spellcheck

Sigil has a built-in spellchecker that is quite powerful, not least because you can add new words to it. In order to add a word to the dictionary, select it and click the “Add To Dictionary” button.

You can also add new replacements, such as “well” for “weU” by editing in the suggestion text box and then clicking “Change Selected Word To:”.

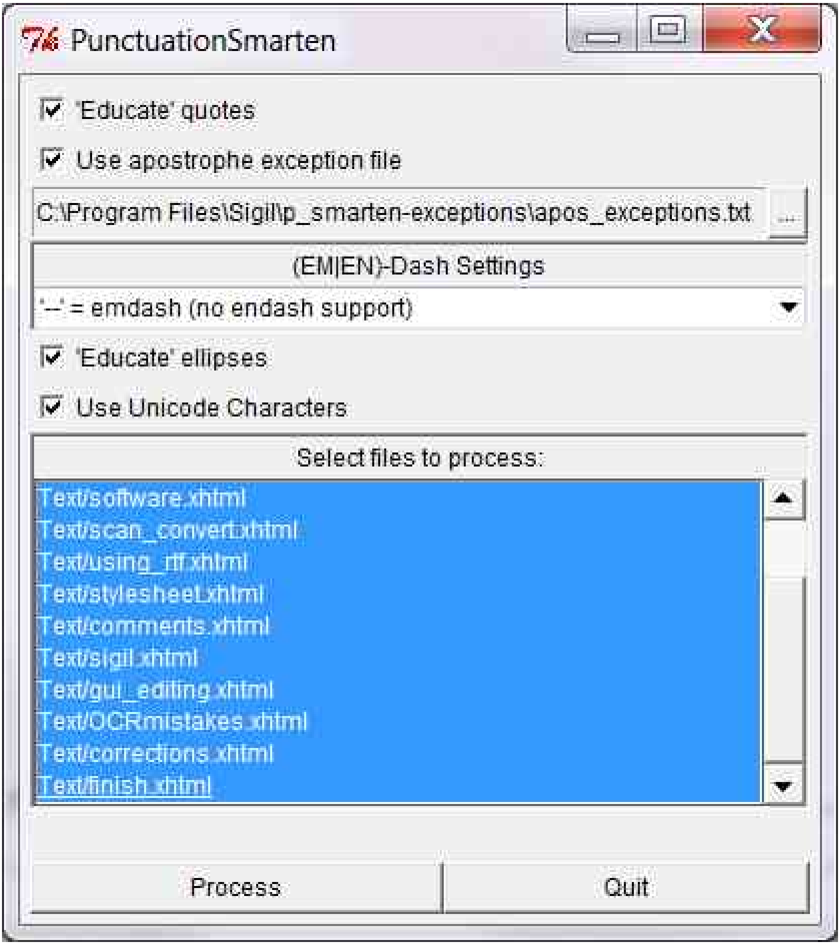
In addition, you can ignore words. This means that the word in question is ignored only for the duration, which is useful for names that may be differently spelled in another book. You don’t want to add a specific spelling to the dictionary. In order to ignore, select the word and click the “Ignore” button.

Apart from the spellchecker, that runs in a little window and lists possibly misspelled words, you can check “Highlight Misspelled Words” in Preferences - Spellcheck Dictionaries. Those words will then be highlighted (underlined with red) in Code View, not Book View, but this still allows you to see them in context, just like you are probably used to from word processors.

Smarten punctuation

Assuming that you have installed the plugin, go to Plugins → “Edit…” → “Punctuation Smarten”, and then click “Start”.

A small windows opens. I leave everything checked there:



PunctuationSmarten

☑ 'Educate' quotes

☑ Use apostrophe exception file

C:\Program Files\Sigil\p_smarten-exceptions\apos_exceptions.txt

(EM|EN)-Dash Settings

'--' = emdash (no endash support)

☑ 'Educate' ellipses

☑ Use Unicode Characters

Select files to process:

```
Text/software.xhtml
Text/scan_convert.xhtml
Text/using_rtf.xhtml
Text/stylesheet.xhtml
Text/comments.xhtml
Text/sigil.xhtml
Text/gui_editing.xhtml
Text/OCRmistakes.xhtml
Text/corrections.xhtml
Text/finish.xhtml
```

Select all files.

Then click “Process”. It usually doesn’t take very long to have all quotation marks smartened. Finish by clicking “OK”.

Next: **corrections**. These will be carried out the same way no matter if you work in Code or Book View. Your EPUB is basically ready now, but it is likely full of OCR mistakes, bad line breaks, and other issues which means that you still have not met those reasonable expectations…

Click the link to the corrections chapter and keep working, or continue on to the next page and read about the Code View workflow…

[*] Now there’s linking back and forth.

Working in Sigil (Code View)

Introduction

Below are good practices to keep in mind all the time. Notably, saving searches and clips saves a lot of work, while saving a copy at the right time might save you from disaster.

Save a copy

There is an option to save a copy in the File menu. Use it before making thorough changes, such as using Replace All when correcting problems. Such a copy may prove very valuable if the result turns out to be undesirable, but you noticed too late.

Searching

Go to Search → “Find & replace…” in order to open the searchbox (at the bottom of the window).

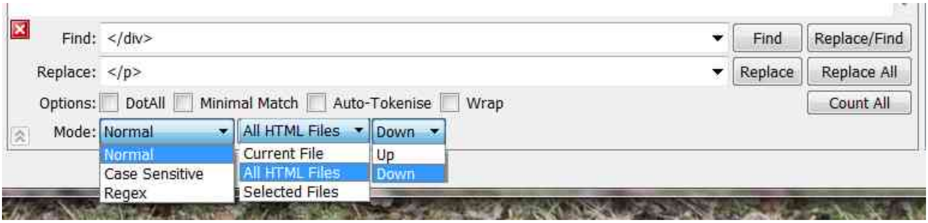
When you want to search the CSS or apply replacements to one page only, make sure that you have selected **Current File**. When you want to search the whole book, select **All HTML Files**.

When you search with **Regex**, it simply will not work unless you have set the search mode to Regex. Normal searches will usually work in Regex, too, unless your search terms happen to include any of the “special characters” that have specific meanings in Regex. Best to switch to Normal mode for normal searches!

Normal mode (or **Case Sensitive**) is “only” for the text as you see it in Code View.

If you stay with the default **Down** direction of searches, make sure that you start your search at the top of the page when searching Current File, and at the very start of the book when you search All HTML Files.

This is an illustrative fake image: you can’t have all the selections showing at once without faking it:



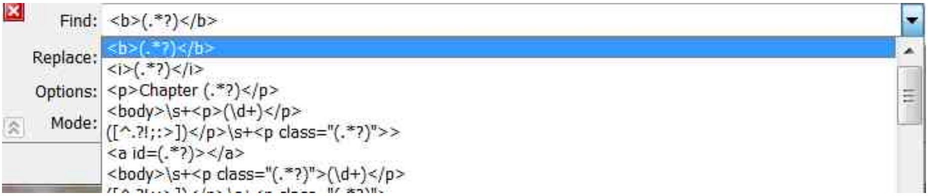
Find: </div>

Replace: </p>

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Normal | All HTML Files | Down

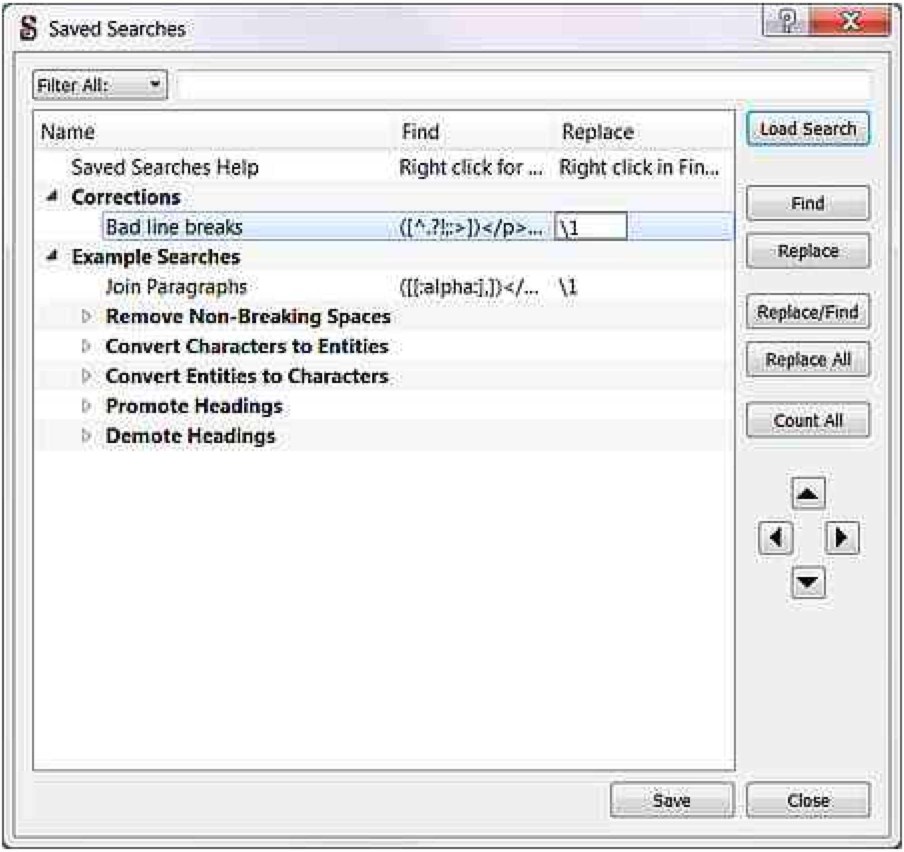
The latest searches are saved in reverse order of appearance (latest saved searches first). This is of course very useful for searches that you repeat many times: just click the little down arrow to the right of the text field (either “Find” or “Replace”) and select the search term you want to use.



Find: <b>(.*?)</b>

```
<b>(.*?)</b>
<i>(.*?)</i>
<p>Chapter (.*?)</p>
<body>\s+<p>(\d+)</p>
([^.?!;:>])</p>\s+<p class="(.*?)">>
<a id=(.*?)></a>
<body>\s+<p class="(.*?)">(\d+)</p>
```

You can also add searches by going to Tools → “Saved Searches…”. There are a number of pre-defined searches there already. You can add and delete Groups and Entries by right-clicking in the window. Here I have added a “Corrections” group, and a search for bad line breaks:



Name	Find	Replace
Saved Searches Help	Right click for ...	Right click in Fin...
Corrections		
Bad line breaks	([^.?!:;>])</p>...	\1
Example Searches		
Join Paragraphs	([[:alpha:],])</...	\1
Remove Non-Breaking Spaces		
Convert Characters to Entities		
Convert Entities to Characters		
Promote Headings		
Demote Headings		

The searches are stored in a file called “searches.ini” or “sigil_searches.ini”, that you can copy and paste into the Sigil directory for use on another computer; the location varies with OSes and versions, but the ini file seems to work everywhere there is a Sigil.

Saving useful text to paste

There is a **Clip Editor** under “Tools” where you can store often-used phrases. Right-click to add Groups and/or Entries, the same way as in Saved Searches. You can access your clips (and useful example ones) with a right-click in the text you are working on.

The clips are stored in a file called “clips.ini” or “sigil_clips.ini”, that you can copy and paste into the Sigil directory for use on another computer; the location varies with OSes and versions, but the ini file seems to work everywhere there is a Sigil.

Cleaning up code

Later versions of Sigil will apparently not prettify the code if you edit something in Book View. Paragraphs tend to run into one another, sharing the same line. It does not matter for the proper functioning of the EPUB, but it makes it difficult to inspect the code.

So I think it is a good idea to run a search now and then (Regex mode):

Find:

```
</p><p class="(.*?)">
```

Replace with:

```
</p>\n\n<p class="\1">
```

This will add a blank line between the paragraphs that were on the same line.

Preparation

What follows are detailed instructions for what to do if you work according to the previously outlined workflow. The next three chapters “only” deal with structure and design, though. The last parts of the workflow, corrections and finishing touches, have got their own chapters.

Copy and edit cover and images

With the PDF open in ABBYY (or other PDF software), copy the cover and save the image in Paint (or other program that allows you to paste and then save in bmp, tif, or possibly as png). Open it and fix as necessary in Photoshop, GIMP, or other similar software. Crop the cover if needed, change brightness/contrast if needed, do other fixes as needed, then save in the original format (such as png, bmp or tif). If the cover is in too poor shape, leave it as it is or go hunting the webs for a better image. N.B. if you need to replace colours, for example in a map that has a brown or grey background that should be white, I recommend leaving it till the IrfanView stage (below); in my experience, it works better than Photoshop.

Sometimes the cover image of a PDF downloaded from Gutenberg or similar repository will be impossible to copy properly (maybe because of the PDF created in A/archive format). The simplest work-around that I have found is to open the PDF in Photoshop, and then work with the cover image from there. It should be possible to open it in GIMP as well.

Also copy the title page if it looks nice enough to insert as a fullpage image instead of making a plain text title page. Edit in Photoshop and then replace background colour as needed using IrfanView.

In addition, copy other images, if there are any, such as maps or small illustrations, and edit as detailed above and below. The size of images that are smaller than fullpage will of course depend on the context; it’s a bit of trial and error, so make sure that you keep the original.

Once edited in Photoshop (or GIMP, etc.), open the image(s) in IrfanView. IrfanView is much better for downscaling with decent quality than Photoshop or GIMP.

If you need to replace colours, open Image - Replace Color and click on the color in the image that you need to replace. It will be automatically added in the source colour field. Then select the new colour. You may need to toy with the tolerance value a bit in order to have the best possible result.

If applicable, then click Image - Convert to Grayscale.

Finally, click Image-Resize/Resample and resize to something suitable – I usually downsize to a height about 750 pixels. Make sure that the option to preserve aspect ratio is checked.

Save in jpg if the image is photolike, with shades and nuances, in gif if there are clearly contrasted full-colour fields. The quality can be low, about 30-60 of 100 or so; this takes the file size down.

Open your EPUB in Sigil

Note that you can change between Book View and Code View by clicking the open book and < > buttons above if using an old version of Sigil, or else open a XHTML file in Page Edit by clicking the square icon that appears to be crossed by a pen.

Add your custom CSS

Use one from the Style Sheets chapter here, or create your own. My instructions will take my style sheets for granted, so you have to re-interpret for your own style sheet, or your edits of mine, when reading this text.

Open the “Styles” folder and double-click “main.css” to open it. Select all the text there and delete it. Copy the text of one of the style sheets here, and paste it into the css file. Save (File → “Save”, or Ctrl+S, just like any word processor).

> *The CSS from ABBY holds very little information; the formatting is done in the text. I replace it all with CSS classes, so that practically no styling is done in the XHTML files. See the Corrections chapter for common replacements — these can be done at any time, so no need to rush over there now!*

However, if you are not working with an EPUB created by exporting from ABBYY, but one created some other way (or downloaded as EPUB), you have to check the style sheet first. **If the italics in the book depend on that style sheet, removing it will cause a small disaster.**

In such cases, you need to search the style sheet for “italic” settings, and copy those classes to your own style sheet. (Personally, I would search for the classes that have “italic” in them in the XHTML files, remove the span styles, and instead apply italics using my style sheet class, or add tags by clicking in Sigil. And then remove those classes from the style sheet.)

Search and replace all &#160; and all &nbsp; with ordinary blank spaces

ABBYY makes lots of them. IMHO it's OK to use non-breaking spaces as needed while designing, but these are superfluous, so get rid of them first.

If the search box is not open, go to the Search menu and click "Find and replace…". Use normal mode and search all XHTML files. Assuming that the default direction "Down" is used, place your cursor at the very first line of the book before you start.

Find	Replace with:

&nbsp;
or
&#160;
	a blank space — **it is very important not to forget this**, since no space here means words running together…

Remove HTML ToC in text file and generate ToC

Look for a table of contents with links that is the result of a ToC in the paper book and remove it, if you find one. *(Yeah, yeah, I know. It would be hard to remove a nonexisting ToC, but oh well. ABBYY will not always insert ToCs, so the question may arise.*) Search for <ul> if you can't spot it right away.

Generate a ToC instead, either by clicking the icon somewhat top right, or by going to Tools → "Table of Contents" → "Generate Table of Contents". Do this right away (and then again as needed, and again when the book is finished); often ABBYY will have added <h> tags that don't show in the ToC when the book is first opened after conversion.

If you are creating a EPUB3, update the toc.ncx for EPUB2 compatibility by clicking Tools - Epub3 Tools - Generate NCX from Nav.

The nav.xhtml file is a different matter. It should be here, do not remove it!

Create main structure

Zoom out the PDF for easy scrolling and **scroll through the PDF and EPUB side-by-side** as it were.

Split XHTML files

If an introductory page has not got its own XHTML file (as it ideally should), you can easily split the file(s): place the cursor at the right place, then hit Ctrl + Enter. These may be title page, copyright page, dedication, epigraph, and others. One file each!

Create a file for the text title page even if you plan to add a large title page image later on. This way, you will have a page in which to insert that image.

(You can also place markers and then perform numerous splits in one go. Place the cursor at the right place, then go to Insert → "Split Marker". When all the markers are in place, go to Edit → "Split At Markers".)

If there are extra pages at the end of the book, such as booklists and back cover text, split them into their own files, too.

Don't worry if the names of the new XHTML files are not pretty. The names don't matter — though I personally usually rename the special pages for easy overview.

Index

If there is an index and the OCR has left it too messy, I simply delete it. (N.B. Nuance often performs better than ABBYY on indexes, lists and tables, so if you have access to it it might be worth OCRing such pages with it, if ABBYY has made a mess.) If it looks OK or needs very little editing, I leave it, but make no effort to make it "live" by linking. I leave it in order to show what the author found worth indexing.

Split XHTML files as needed, one for each chapter

Split larger files resulting from the conversion. Search in the EPUB for an unusual word near the chapter beginning in order to quickly find it.

Split the same way as you did for the introductory "special" pages: place the cursor at the right place, then hit Ctrl + Enter. Don't worry if the names of the new XHTML files are not pretty. The names don't matter.

An EPUB will work even if you don't bother to have one XHTML file per chapter (and one XHTML file per special page), but apart from being neat, it also makes conversions to MOBI or AZW better, since they tend to "look" at XHTML files.

> *Optional (if you are orderly, like me:) if there are orphaned XHTML files due to splitting at chapters, copy and paste the text into the correct chapter XHTML file, then delete the superfluous files. One XHTML file for each chapter.*

Add cover

Go to Tools → "Add cover..." and browse to your saved (jpg or gif) image. Sigil will create a new cover.xhtml file, placing it first in the list of XHTML files.

N.B. if your cover isn't already called "cover", right-click the image in the Image folder(in the Book Browser to the left) and rename it. It will not show properly in Calibre (and some other software) if it is not called "cover.jpg" (or, cover.png, or cover.gif, etc.)

Add a description; see "Fullpage images" below.

Add XHTML files

Sometimes (rarely), you will need to add XHTML files, for example in order to insert a fullpage map into the middle of the book.

Right-click the XHTML file above the desired place of the new page, and choose "Add blank HTML file". It will be called "section-something" and there's no need to rename it. It is the order of files, as shown in the left-hand book browser, that is important.

The new file will have a very basic head section. Notably, the link to the CSS file will not be copied, and the result is of course that your CSS has no effect there. So, go to an "old" page and copy the head section of the XHTML file. Typically it will look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns:epub="http://www.idpf.org/2007/ops">
<head>
  <title>Pride and Such</title>
  <link href="../Styles/main.css" rel="stylesheet" type="text/css"/>
</head>
```

Paste it into the new page (in Code View), replacing the old head section.

See “Fullpage images (large maps, copied title page, etc.)” for how to insert such images.

More images

If you spot any **images** that you haven’t already copied, copy them from the PDF, fix and rescale them, so that you can insert them in the EPUB where they should be – and insert any already copied and fixed images you saved earlier.

You insert images by going to Insert → “File…” and then clicking the “Other files…” button. Browse to where you have saved the image to be added. If you have downscaled and saved as jpg or gif as instructed, make sure that you don’t accidentally add the original, needlessly large, one instead.

Do remember to add a short and relevant description to the “alt” attribute inside the< img > tag, like this:

```
<figure>
  <p class="text"><img alt="Image of a red dragon head, with open mouth
breathing fire" src="../Images/red-dragon.jpg"/></p>
</figure>
```

Fullpage images (copied title page, large map, etc.)

In order to make fullpage images cover a full page, while keeping correct ratio and not overflow (turn larger than the e-reader can show), you need to go into Code View and apply some changes to a (new) blank XHTML file.

Use Insert - File... first, inserting the image in its own otherwise blank XHTML file, so that the image is added to the book, then edit the page. If you don’t know the size of the image, open the Images folder in the Book Browser pane to the left and double-click the image file name; the information is there.

A cover added using the Add Cover tool is marked up like this; I use it for other images that need to be full-page, too, just changing the measurements (two places each, width and height in the “viewBox”, then height and width for the image) and of course the file name. The bold text shows what will need to be changed.

```
<div style="text-align: center; padding: 0pt; margin: 0pt;">
```

```
    <svg xmlns="http://www.w3.org/2000/svg" height="100%"
preserveAspectRatio="xMidYMid meet" version="1.1" viewBox="0 0 431 750"
width="100%" xmlns:xlink="http://www.w3.org/1999/xlink">
      <image width="431" height="750" xlink:href="../Images/cover.jpg"/>
    </svg>
  </div>
```

In addition, add a description using a <desc> tag. This is a work-around since you cannot use the “alt” attribute here. Something like this:

```
<div style="text-align: center; padding: 0pt; margin: 0pt;">
    <svg xmlns="http://www.w3.org/2000/svg" height="100%"
preserveAspectRatio="xMidYMid meet" version="1.1" viewBox="0 0 431 750"
width="100%" xmlns:xlink="http://www.w3.org/1999/xlink">
      <image width="431" height="750" xlink:href="../Images/cover.jpg"/>
<desc>A map showing islands in the sea, and a warning for dragons at the
border of the map</desc>
    </svg>
  </div>
```

Design details

The workflow below is approximate, as it has to be: you will find different items to design in different books, and restructuring the order of the workflow will sometimes make it all easier…

Special pages design

I don’t try to mimic the paper book exactly, but will use classes for copyright (small), halftitle, author, booktitle, and publisher, and also for dedication, etc. as needed.

When these are all on their own respective XHTML files, it is easy to find all <p> and replace with the appropriate class, such as <p class="copyright"> for the copyright page. **N.B. it is very important to have the Find and replace box set to Current file when performing these replacements**. You don’t want the whole book formatted in the smallish copyright style.

In some cases you can’t simply search and replace all <p> tags in the current file. Author, title and publisher for the title page will have to be added manually. I insert “-fl” manually for blank lines at the copyright page — you can simply add a blank line instead. Sometimes there is an epigraph with the source on the line below; I usually set that source line to <p class="right"> manually.

Add the semantic tags at this stage, too. Replace the <body> tag with <body epub:type="frontmatter"> for the first batch of special pages, and use <body epub:type="backmatter"> if there are special pages at the end of the book.

Additionally, add sections with appropriate EPUB types; there is a list in probably book order in Appendix 2. Remember to close with </section> at the bottom of the page, before the </body> tag.

Make the chapter headings h2 or h3

Select text for a heading, then click one of the h1-h6 buttons. Perfect markup, much easier than adding <h2> (etc.) tags manually.

Then either add sections with appropriate EPUB types (usually chapters) manually, or run a couple of Regex searches:

Find:	Replace with:

<body epub:type="bodymatter">\n<section epub:type="chapter">	
and then find: </body> *and then replace with:* </section>\n</body>	

If there are other types than chapters to be added to <section> tags, such as an epilogue, go back and change to the appropriate EPUB type manually.

Generate ToC again

Do this when all the chapters have got their headings (and their own XHTML files), and then check that all the chapters are there. You can add or remove items in the ToC, and also edit the entries by going to Tools → “Table of Contents” → “Edit Table of Contents...”. This is useful when the chapter headings are so long that the ToC becomes hard to read.

Remember to generate NCX from Nav if you are creating an EPUB3.

Replace span styles with <em>, <i>, and <b>

The common occurrences are easy to deal with, though I would recommend running at least the italic search manually in order to apply <em> or <i> tags correctly.

> *I just went through a book with lots of book and magazine/paper names, and plenty of foreign expressions; most in French, but a few in German and Latin, too. I ran a search-and-replace manually for <span style="font-style:italic;">(.*?)</span> with replacement <em>\1</em>, but manually changed to <i>[italic-word]</i> instead when appropriate. In addition, I copied the foreign words to a text file, and then ran a second search for those words, changing the <i> to <i xml:lang="de">, <i xml:lang="fr">, and <i xml:lang="la">, respectively. Then I ran a search for the remaining <i>(.*?)</i> expressions, which were names of books or magazines/papers, and changed them all to <cite>\1</cite>. It took a while, yes, but it is now all correct.*
>
> *N.B. the strategy would have to be different for a book with lots of “alternate voices”, like letters and songs, which should have <i> tags only. Maybe paste <cite> tags while doing the first search intending to replace with <em> where appropriate. The bottom line is that you will have to adapt different strategies for different books.*
>
> *To be honest, when there are hundreds of italic expressions, I tend to replace all with <em>, and then fix <i> and <cite> when proofreading, possibly with some search-and-replace if some word or name is repeated several times.*

Find:	Replace with:

<span style="font-style:italic;">(.*?)</span>	<em>\1</em> or possibly <i>\1</i>

and

Find:	Replace with:
<span style="font-weight:bold;">(.*?)</span>	<b>\1</b>

N.B. bold style is often mistakenly applied when ABBYY for unknown reasons has interpreted normal-weighted text as bold, so you might want to run the bold search manually, too.

There may also be mixed cases, which should usually only be italics <i>, but possibly <em>, of course. Add <b> or <strong> tags later, if they are called for after all.

Find:	Replace with:
<span style="font-weight:bold;font-style:italic;">(.*?)</span>	<i>\1</i> or possibly <em>\1</em>

ABBYY will also commit atrocities such as:

<span style="font-weight:bold;">V. L</span> <span style="font-weight:bold;font-style:italic;">f</span> <span style="font-weight:bold;">FADING</span>

It is actually easier to deal with these span combinations by clicking the buttons for bold or italic text than coming up with search terms to catch them all, or to edit out the spans in Code View. You will sometimes need to click twice to get rid of bold and/or italic styles. Just check with the PDF what it should be, if it is not obvious.

Special styles such as letters and poems

If and when you spot any deviant items, such as **telegrams, letters, poems, and songs**, style them so that they stand out from the running text, similar to but not necessarily the exact same way as they do in the PDF. The extended style sheet contains classes for “letter” and “poem”, that should also work OK for telegrams and songs, respectively (if you do not wish to add your own new classes). **Remember to add <i> tags for these items so that they are marked up as being in an “alternate voice”.** N.B. you can of course just add the classes first, and then run a search for <p class="poem>(.*?)</p> and replace with <p

class="poem"><i>\1</i></p>, etc. That saves a little bit of typing.

Smallcaps

If there are SMALLCAPS, just leave them be if they are at the start of chapters. They will not be parsed by all e-readers, but that will do no harm there. If there are smallcaps in the running text, for signs or telegrams or newspaper headlines, etc., remove the smallcaps span style/class, then select the words that should be in caps and click the “AB” button somewhat to the right above. That will turn all the letters into UPPER CASE.

> *If you are particular like me, you will change those words to <span class="small">UPPER CASE</span>, but it is perfectly fine to just leave the uppercase letters as they are.*

Add scenechange lines

...whenever you see a scenechange indicated in the PDF. **This is important**: abrupt changes with no indication makes for a poor reading experience. So, indicate scenechanges by adding a line with some visual clue such as

```
<p class="scenebreak">* * *</p>
```

(and make the class go centered in the CSS). The commonly used blank line <p><br /></p> will not always be correctly parsed.

It will look nice if you also add <p class="firstline"> instead of <p class="text"> after the scenebreak, with no indent as opposed to the normal <p class="text"> (and perhaps a small top margin set for the firstline class in CSS to boot).

Footnotes

Some novels contain **footnotes**, though usually not as many as there can be in text books. These, that are originally usually placed at the bottom of a paper page, go to the end of the chapter (XHTML file), with linking back and forth. The footnote text is placed inside <aside> tags in order to tell a screen reader that it is not part of the running text.

The reference in the text:

```
<span class="ref"><a epub:type="noteref" id="back1" href="#note1">1</
a></span>
```

The footnote text (placed at the very end of the chapter):

```
<aside epub:type="footnote">
<p class="footnote"><a epub:type="backlink" id="note1" href="#back1">1
</a>[FOOTNOTE TEXT]</p>
</aside>
```

You may want to add a line (just press the __ key) above the footnote text, making it visually clear that it is not part of the running text.

Replace <p> with <p class="text">

Now everything that shouldn’t be running text is tagged differently; replace <p> with <p class="text">. You can leave it <p> and add margin:0 (and possibly indent:1.2em or so) to the style sheet for “p” if you wish, but rumour has it that some e-readers will not parse CSS for classless <p>, so in that case there will be spaces between all paragraphs on those readers.

Unexpected and unwanted markup when editing in Book View

A lazy Code View worker will occasionally switch to Book View. However, this will sometimes lead to unwanted changes in the design. Sigil reads the style sheet settings and may apply them as styles in the text when you remove lines or move parts of the text.

You may end up with a heading going larger than the other ones of the same level because Sigil added a 1.2em span style to the existing 1.2em font-size in the style sheet, or a heading getting a paragraph class when moved up by deleting previous lines, or indeed a heading changed into a paragraph with the heading attributes set in the shape of span styles. This is extra problematic since that (previous) heading will not be included when you generate a new ToC.

So it is a good idea to run a search for the catch-all “style=” and for “h2 class=” and “h3 class=” (etc. if you use more classes).

> *You can add styles at three locations: inline, i.e. in the text of the XHTML file (along with the HTML tags), embedded at the top of the XHTML file, and in an external style sheet (a CSS file) that the XHTML file links to. Styles in the text of the XHTML file override styles at the top of the page as well as styles in a style sheet/CSS file. (Styles at the top of the page override styles in a style sheet/CSS, but not those in the text.) So: styles in the text of the XHTML file can mess up what you want to accomplish with a style sheet.*

Also, I have noticed that Sigil will sometimes create a <div> instead of a <p> when you add a new line by pressing Enter in Book View. These <div>s will of course not be affected by the style settings in your CSS, so you want to change them to <p>. Run a search for <div>, and change them to <p> (or <p class="text">, and then </div> to </p> — if you know that you haven't used <div> in the book for some legitimate purpose. N.B you need to run both searches before saving, or Sigil will complain about errors and offer to fix them in unforeseeable ways.

In addition, blank spaces may turn into &#160;. I try to remember to check the text if I have edited in Book View, getting rid of superfluous non-breaking spaces again.

Spellcheck

Sigil has a built-in spellchecker that is quite powerful, not least because you can add new words to it. In order to add a word to the dictionary, select it and click the "Add To Dictionary" button.

You can also add new replacements, such as "well" for "weU" by editing in the suggestion text box and then clicking "Change Selected Word To:".

In addition, you can ignore words. This means that the word in question is ignored only for the duration, which is useful for names that may be differently spelled in another book. You don't want to add a specific spelling to the dictionary. In order to ignore, select the word and click the "Ignore" button.

Apart from the spellchecker, that runs in a little window and lists possibly misspelled words, you can check "Highlight Misspelled Words" in Preferences - Spellcheck Dictionaries. Those words will then be highlighted (underlined with red) in Code View, not Book View, but this still allows you to see them in context, just like you are probably used to from word processors.

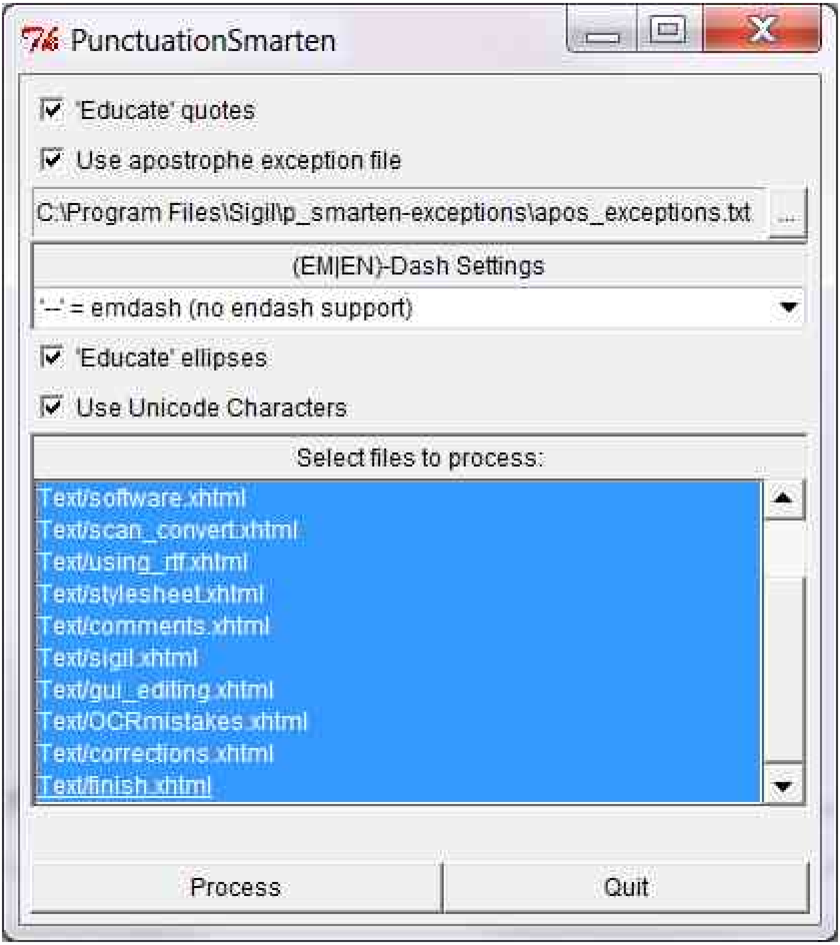
If you prefer to see misspelled words highlighted in something like Book View, you might consider installing the later version of Sigil, and also install the new PageEdit program and "link" it to Sigil. N.B. that the Book View is gone from later Sigil versions, so you will have to do with a small preview pan when not opening a XHTML file in PageEdit.

PageEdit allows you to check spelling the well-known reddish underline way of many word processors, without seeing the underlying code. The drawback is that you can't "train" the dictionary by adding new words.

Smarten punctuation

Assuming that you have installed the plugin, go to Plugins → "Edit…" → "Punctuation Smarten", and then click "Start".

A new small windows opens. I leave everything checked there:



PunctuationSmarten

☑ 'Educate' quotes

☑ Use apostrophe exception file

C:\Program Files\Sigil\p_smarten-exceptions\apos_exceptions.txt

(EM|EN)-Dash Settings

'--' = emdash (no endash support)

☑ 'Educate' ellipses

☑ Use Unicode Characters

Select files to process:

Text/software.xhtml
Text/scan_convert.xhtml
Text/using_rtf.xhtml
Text/stylesheet.xhtml
Text/comments.xhtml
Text/sigil.xhtml
Text/gui_editing.xhtml
Text/OCRmistakes.xhtml
Text/corrections.xhtml
Text/finish.xhtml

Select all files.

Then click "Process". It usually doesn't take very long to have all quotation marks smartened. Finish by clicking "OK".

Next: corrections

These will be carried out the same way no matter if you work in Code or Book View. Your EPUB is basically ready now, but it is likely full of OCR mistakes, bad line breaks, and other issues which means that you still have not met those reasonable expectations…

Corrections and Fixes

Note: "BV" indicates if you should do the search if you work in Book View, while "CV" indicates that you should run the search if you work in Code View. Obviously, some searches should be run in both cases.

Some CV searches, such as adding firstline, are clearly optional.

You want to have as many problems as possible corrected by searches before it's time to proofread your book. It saves a lot of time not having to break off reading and perform manual corrections all the time.

If you are creating many EPUBs, it may be a good idea to add the searches in "Saved Searches", where you can store "Find" and "Replace" together. You can right-click in the Searchbox and save the current search from there, and also load saved searches by right-clicking in the box.

> *When I occasionally say "run manually" below, it is usually short for: "Click Find, check the result, and if it should be replaced with the option in the Replace box, click Replace (or Replace/Find for immediate search for the next instance), but don't click Replace All. If the result shouldn't be changed at all, click Find again. If it should be replaced with something other than what's in the Replace box, edit it manually (for real)."*

Using Regex mode

These are searches that should be run on a newly designed EPUB, but the list is not, and cannot be, complete. There will practically always be some unforeseen problem. You will probably spot it as you proofread and so able to correct it by editing the text, but if you create many EPUBs it is well worth while to learn how to write your own Regex searches in order to deal with repeat problems. N.B. there are different "flavours" of Regex, so if you follow some online course or guide, make sure that it is the right "flavour" by testing the given examples in Sigil.

Missing chapter heading tags (BV & CV)

For cases of <p> instead of <h2> (or <h3>, etc., change to suit you). Of course there shouldn't be any of these if you have designed the book properly, but you never know… Also, this one is useful if you edit a downloaded EPUB.

To use when there is a pattern (such as "Chapter…"):

Find:	Replace with:
<p>Chapter (.*?)</p>	<h2>Chapter \1</h2>

To use when there are only chapter numbers. This will find chapter numbers only, provided that there is nothing else between the <body> tag and the chapter number. You may have to remove anchors first (see below):

Find:	Replace with:
<body>\s*<p>(\d+)</p> or <body>\s*<p class="(.*?)">(\d+)</p>	<h2>\1</h2> or <h2>\2</h2>

Page numbers remaining after OCR (BV & CV)

(This one is useful if you edit a downloaded EPUB, too.)

To use when there is a pattern (such as “Page…”) — beware of “Page…” mentioned in the text, though, so don’t replace all at once.

Find:	Replace with:
<p class="(.*?)">Page \d+</p> or <p>Page \d+</p>	nothing (blank “Replace” box)

To use when there are only page numbers. N.B. if there are chapter numbers that should be in <h> tags in <p> or <p class="text">tags, fix them before doing this (see “missing chapter heading tags” above).

Find:	Replace with:
<p class="(.*?)">\d+</p> or <p>\d+</p>	nothing (blank “Replace” box)

Bad line breaks (BV & CV)

The first search here is an update (due to a clever question from dearleuk) about searching for paragraphs that begin with a lower case letter instead of just searching for paragraphs that don’t end with a proper sentence-closing character. I would recommend first searching for these lower case paragraphs, and would indeed be tempted to replace them all in one go: paragraphs that legitimately start with a lower case letter will be extremely rare.

Afterwards, run the second search for paragraphs ending with presumably wrong characters: that one will find missing period marks along with various OCR artefacts, and should still be run “manually”. There will be fewer hits when all the lower case lines have already been found, and so the correction work will be faster.

1. Paragraphs that begin with a lower case letter. **N.B. a space before the replacement.**

Find:	Replace with:
</p>\s+<p class="(.*?)">([a-z]\w*\s) or </p>\s+<p>([a-z]\w*\s)	\2 or \1 *N.B. the space before the replacement*

2. A catch-all that finds paragraphs that don’t end with period (full stop), question mark, exclamation mark, semi-colon, colon, or the last part of a closing span tag. Check one by one, sometimes there is just a period mark/ full stop missing at the end of a paragraph rather than a bad line break, or it could end with em dash… **N.B. replacement followed by a space.**

You may need to add ” or ’ inside the square bracket for double or single quote marks.

Find:	Replace with:
([^.?!;:>])</p>\s+<p class="(.*?)"> or ([^.?!;:>])</p>\s+<p>	\1 *N.B. the space after “\1”*

Missing line breaks in conversations (BV & CV)

For double main quotes. Run manually (since some conversations do not have linebreaks):

Find:	Replace with:
”</p>\n\n<p class="(.*?)">“ or ”</p>\n\n<p>“	”(.)“

For single main quotes. Run manually (since some conversations do not have linebreaks):

Find:	Replace with:

’</p>\n\n<p class="(.*?)">‘ or ’</p>\n\n<p>‘	(.)

Remove incorrect line breaks due to leftover hyphens (BV & CV)

Can probably be run all at once, though there *may* be some rare occasions when a paragraph legitimately ends with a hyphen, and the next paragraph starts with a lower case letter.

Find:	Replace with:
([a-z])-</p>\s+<p class="(.*?)">([a-z]) or ([a-z])-</p>\s+<p>([a-z])	\1\2

Remove <sup> and <sub> tags (BV & CV)

ABBYY may add some <sup> or <sub> tags, lifting up or sinking words and letters for seemingly no reason. If you have no legitimate <sup> tags in your book (i.e. haven’t used them for footnotes/links), you can remove all the <sup>s in one go, using Regex. If you have intentionally used <sup>, you will of course have to check the search results manually.

Find:	Replace with:
<sup>(.*?)</sup> and then <sub>(.*?)</sub>	\1 (or nothing)

Remove anchors (BV & CV)

- such as <a id="bookmark0"></a> in which the id changes (to "bookmark1", "referenceX", etc). If you remove the HTML ToC there will often still be targets (anchors) left in the text. Remove them this way:

Find:	Replace with:
<a id=(.*?)></a>	nothing

Remove hyperlinks (BV & CV)

- such as <a href="http://address.org">Text here</a>, when there are several links with different URLs and you want to keep the text ("Text here") without live linking.

Find:	Replace with:
<a href=(.*?)>(.*?)</a>	\2

Remove mistaken underlining (BV & CV)

Sometimes ABBYY will underline words such as "him" and "think" that may seem underlined to the OCR, especially with serif fonts (which are used for most novels).

Find:	Replace with:
<span style="text-decoration:underline;">(.*?)</span>	\1

"l" (ell) instead of exclamation mark "!" (BV & CV)

Since exclamation marks are usually at the end of a sentence, search for "l" with lower case letters before and a space or a right-side quotation mark after. There will be a lot of suggested searches since quotation marks can be differently coded, and the space may be an ordinary one, or a &nbsp; or a &#160; — and then there may be span tags…

Inside a paragraph: a sentence that should end with an exclamation mark, followed by a new sentence starting with an upper case letter. (The ^/ addition is so that the top of the XHTML file is ignored.) **This will find several words legitimately ending with l followed by, usually, a name, too.** Replace manually by editing in Code View, or use ! \1, always checking first that the word doesn't legitimately end with l:

Find:	Replace with:
[l](\s[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)] or [l](&#160;[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)]	!\1

At the end of a paragraph. You can replace with !</p>, provided that you have already searched for bad line breaks:

Find:	Replace with:
[l]</p>	!</p>

At the end of utterances ending a paragraph. You can replace with !"</p> or !'</p> according to the quotation mark coding used in the text:

Find:	Replace with:
[l]”</p> or [l]’</p>	!”</p> or !’</p>

If you wish to continue and search also for styled sentences that should end with an exclamation mark, add <span>, </b> or </i> after [l].

“P” instead of question mark “?” (BV & CV)

Same reasoning as for exclamation marks since question marks also usually appear at the end of sentences.

In a paragraph: sentence that should end with an question mark followed by a new sentence starting with an upper case letter — replace manually by editing in Code View, or use “?\1”, always checking first that the word doesn’t legitimately end with upper-case P:

Find:	Replace with:
[P](\s[A-Z]) or [P](&#160;[A-Z])	?\1

At the end of a paragraph; you can replace with “?</p>”, provided that you have already searched for bad line breaks:

Find:	Replace with:
[P]</p>	?</p>

At the end of utterances ending a paragraph; you can replace with ?”</p> or ?’</p> according to the quotation mark coding used in the text:

Find:	Replace with:
[P]”</p> or [P]’</p>	?”</p> or ?’</p>

If you wish to continue and search also for styled sentences that should end with an question mark, add <span>, </b> or </i> after [P].

Italic or bold applied to single letter (BV & CV)

May be OCR error, but may also be correct if it’s a one-letter word (“Are you suggesting that *I* did it?”), so replace manually and check PDF when necessary.

Find:	Replace with:
<b>(.?)</b> or	\1

<i>(.?)</i>	

For a different first line following headings (CV)

Run when you are sure that all the chapter headings are in place.

Find:	Replace with:
</h2>\s+<p> and/or </h3>\s+<p>	</h2>\n\n<p class="firstline"> and/or </h3>\n\n<p class="firstline">

For a different first line following scenebreaks/ scenechanges (CV)

Find:	Replace with:
<p class="scenebreak">(.*?)</p>\s+<p class="(.*?)">	<p class="scenebreak">\1</p>\n\n<p class="firstline">

Using Normal mode

Replace non-breaking spaces with HTML entity or numerical.

ABBYY (and other software for converting files to EPUB) has a tendency to add non-breaking spaces where ordinary spaces would be just fine. It is a very good idea to replace them before starting to edit the book, since you may want to add non-breaking spaces in some special circumstances later on.

So, run a search for &nbsp; and for &#160; and replace with a space. Just replace one instance first and make sure that there is a space there. (Yes, this search is already mentioned in the workflow. Repeated here, so that it won't go missing if only search suggestions are copied.)

/ (slash) instead of…

…various things. OCR can mistake italic I or italic ! for a slash, along with phrases ending with a comma or period followed by a single or double quote (*here is an example,”*) . You will catch those that end a paragraph by searching for “/</p>”, and the ones that should be “I” by searching for / with a space before and after. This will have to be done manually since you cannot be sure that all slashes should have the same replacements. And then find and correct the last of them while proofreading.

“1” (one) or “l” (ell) instead of “I” (BV & CV)

You will find most of them by a normal search for “1” with one space before and one after, and “l” with one space before and one after. Add a search for “<p>1” and “<p>l”, and/or “<p class="(.*?)">1” and “<p class="(.*?)">l” (search Case Sensitive for the “l” here), and you will catch the majority. N.B. remember to add a space after “I”.

“rn” instead of “m” and vice versa (BV & CV)

A few examples:

modern - modem

burn - bum

corner - comer

Search for each, or leave it for the proofreading — assuming that you are sure you will spot them then. Don’t click “Replace All” — that could, for example, make modems modern.

“lie” instead of “he” (BV & CV)

Search obvious, of course checking for a true lie in the text each time.

“Pie” or “Fie” instead of “He” or “She” (BV & CV)

Search obvious, of course checking for real Pies and Fies in the text each time.

> *The above is simply a list of some common OCR mistakes that cannot be corrected by running a spellcheck. These mistakes are words and letters that can appear in the book legitimately, or indeed because the OCR software got it wrong. There’s no telling which it is without looking at the context in which they appear.*
>
> *Since the mistaken words and letters may be legitimate, you can’t run an automated Find and Replace All. That might introduce new problems, by (for example) turning true bums into burns… So, you have to check for them “manually”, searching and*

clicking “Replace” only if the word is truly wrong.

The list is quite short: more of an inspiration for you than an attempt to be complete. It really is impossible to create a complete list — I will run searches for known occurrences like the ones above, but will often have to take a break in proofreading because some other OCR mistake has turned up, and seems to be repeated so that a search will save time compared to correcting each instance separately.

Unwanted span styles not previously corrected (CV)

I usually run a last search for “<span style=” with no other specification to find all such instances that ABBYY may have introduced, and may switch to a more specific “Find & Replace All” if there is some common issue, before continuing the unspecific “<span style=” search.

Apply quotefix (CV)

For double main quotes, left side:

Find:	Replace with:
“<span class="quotefix">“&#160;</span>‘ or “&#160;‘	“‘ or “‘

For double main quotes, right side:

Find:	Replace with:
’<span class="quotefix">’&#160;</span>” or ’&#160;”	’” or ’”

For single main quotes, left side:

Find:	Replace with:
‘<span class="quotefix">‘&#160;</span>“ or ‘&#160;“	‘“ or ‘“

For single main quotes, right side:

Find:	Replace with:
”<span class="quotefix">”&#160;</span>’	”’ or ”’

or ”&#160;’	

The exception: Flatten ToC with Calibre e-book editor (BV & CV)

I have come across a few EPUBs with ToCs that were levelled when they shouldn’t be. ToCs are legitimately levelled when there are, for example, a chapter h2 heading and a number of h3 (sub)headings in that chapter. The ToC will show the h3 headings one level down, under the h2 heading.

Then there are those ToCs in which, for example, a h2 is inexplicably placed “under” another h2. It can’t be fixed by correcting the headings, because they are already correct.

This is when the Calibre e-book editor comes in handy — at least if there are no legitimately levelled headings in the book.

After opening the EPUB in the editor, go to Tools → “ToC - Edit Toc”, and then press the “Flatten ToC” button. Save and close the program.

The Finishing Touches

Add your custom CSS (BV)

This is optional for the Book View editor: use a style sheet for applying styles that are not available in the GUI, notably for headings and paragraphs (text-indent for running text in a novel). Use the “Replacement CSS” from the Style Sheets chapter here (or create your own) if you want to have control of the headings, and have running text without blank spaces between the paragraphs and indented like this:

> I bowed. “The highest pleasure of the altruist,” I replied, “is in contemplating the good fortune of others.”
> Mrs. Haldean laughed. “Thank you,” she said. “You are quite unchanged, I perceive. Still as suave and as—shall I say oleaginous?”
> “No, please don’t!” I exclaimed in a tone of alarm.
> “Then I won’t. But what does Dr. Thorndyke say to this backsliding on your part? How does he regard this relapse from medical jurisprudence to common general practice?”

Open the “Styles” folder to the left in Sigil and double-click “main.css” to open it. Copy the text of the “Replacement CSS”, and paste it into the css file. N.B. if you left classes with italics in the style sheet before, don’t remove them now. Paste the “Replacement CSS” in addition to such classes. Save (File → “Save”, or Ctrl + S, just like any word processor).

If you “only” want indented text, but prefer the headings Sigil gave you, remove the h-lines in the style sheet.

Clean up the style sheet (BV & CV)

When the book is finished, I run Tools → “Remove Unused Stylesheet Classes”, which makes any unused special class disappear. It doesn’t happen right away: you will get a preview of what is about to be deleted. Standard classes, such as h2, h3, and td will remain even if they are not used in the XHTML files, so you need to remove them manually from the style page if you want it perfectly slimmed down. Provided that you really did not use them, of course.

Add landmarks (CV)

In addition to adding sections and specifying EPUB types in the XHTML text files, you should also add landmarks to the nav.xhtml file (that will have been created from start when creating an EPUB3, or added when running the ePub3-itizer plugin). They are intended to provide an overview of the structure of the book to a screen reader, so you should only add a few major ones. Typically, landmarks are hidden so that only screen readers (and other assistive technologies) will “see” them.

If you open your nav.xhtml file in Code View, you will probably see it starting with a simple table within <nav epub:type="toc" id="toc"> and </nav> tags.

There may be a <nav epub:type="landmarks" id="landmarks" hidden=""> entry at the end of the file, too - if there is not, create it, and close it with a </nav> tag. Then add the landmarks table in between, copying references to major sections from the ToC entries above.

Most novel landmarks will look something like this, marking the start of the EPUB (<body epub:type="frontmatter"> such as the cover), the start of the actual book (the first file with <body epub:type="bodymatter">) such as Chapter 1, or maybe a prologue), and possibly some backmatter (the first file with <body epub:type="backmatter">) such as book lists or a colophon.

Note that you are free to add text that explains the purpose of the landmark; a reference to the first chapter may be called “Start of book”, etc., and the start of backmatter may indeed be called “Text from dust cover” even if there was no such EPUB type for such content.

```
<nav epub:type="landmarks" hidden="">
    <ol>
        <li><a epub:type="frontmatter" href="../Text/cover.xhtml">Cover</a></li>
        <li><a epub:type="bodymatter" href="../Text/main-1.xhtml">Start of Book</a></li>
        <li><a epub:type="backmatter" href="../Text/otherbooks.xhtml">List of Published Books</a></li>
    </ol>
</nav>
```

Add title to <head>

When ABBYY has converted a PDF to EPUB, there will be an empty <title> tag in the head of the XHTML files. This would cause the EPUBCheck to fail, so you need to add a title.

The simple solution is to search for <title></title> and replace with

<title>[BOOKTITLE]</title>. In the case of this book, it would result in

```
<head>
  <title>E-Books That Cut It, 2nd ed</title>
  <link href="../Styles/main.css" rel="stylesheet" type="text/css"/>
</head>
```

Or, you can go through all the XHTML files and add the heading of each file as the title.

Add metadata if needed

If you converted your PDF from ABBY, you have probably already added Title and Author (Creator) to the EPUB's metadata by entering them as you saved. You want your book to have at least those two pieces of metadata, plus language.

If your EPUB lacks those entires, or if you want to edit them, got to Tools → "Metadata Editor…". Click the Add Metadata button in the Metadata Editor window, and then click Title. This will give you a row for dc:title. Double-click the Value field and submit the book title. Then Click the Add Metadata button again, this time clicking Creator, which will give you a row called dc:creator for the main author of the book. You can add more authors by repeating the process. Also add Language (dc:language) if it hasn't been added automatically.

Validate with EPUBCheck (BV & CV)

Normally there will be nothing to correct here — but if the check shows some error, you will of course have to fix it. A successful run of EPUBCheck assures that the EPUB is created in accordance with the EPUB standard, structured as it should be, and contains everything it should (in addition to the text itself), so errors might mean that your EPUB will not work as intended when you try to read on your e-reader.

Run Access-Aide (CV)

This will automatically update some accessibility features, including adding an accessibility schema to the content.opf file and presenting all images in a way that makes it easy to edit the alt attribute.

Proofread (BV & CV)

Yes, it takes time and is often boring, compared to just reading, but it is worth it for the quality gain. There are usually lots of little OCR mistakes not previously detected.

I used to proofread using my e-reader, getting up all the time for performing corrections in Sigil on the computer. *That* was boring! Then I got myself a small laptop, installed Sigil, and have since proofread using that one. I copy the searchable PDF to the laptop, too, so that I can quickly check if I am unsure about whether something really is correct or not.

One thing to look out for, apart from all the possible ordinary issues: ABBYY occasionally discards whole chunks of text, typically the paragraph(s) that follow a heading, an image, or a header. Most of the time you catch it when searching for bad line breaks. But without carefully reading the EPUB against the PDF, there is no way to be sure that the EPUB really contains all the text of the original…

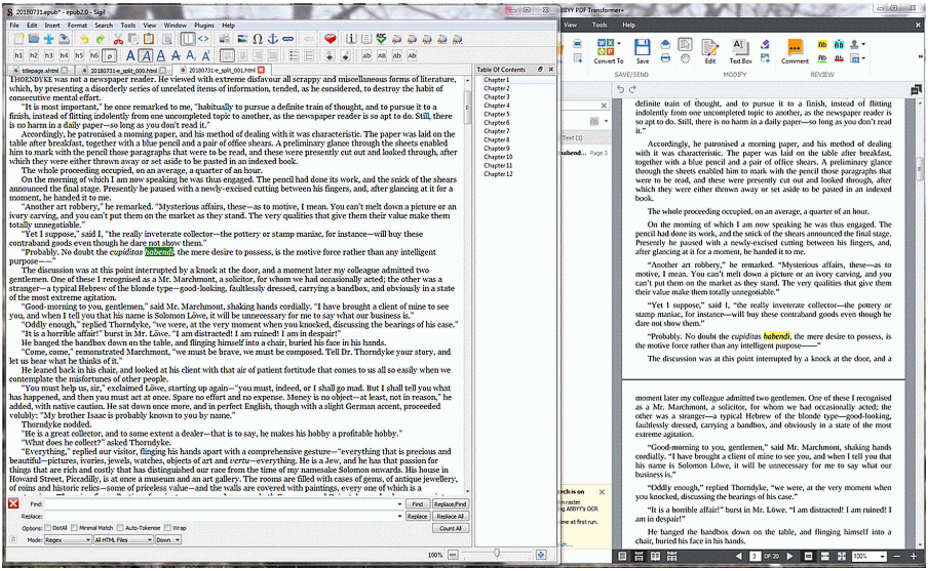
I am not enough of a perfectionist to do that comparative reading, and I suspect that none or few others are, either. But: if you notice a "jump" in the text, do go back to the PDF and check if there really is something missing.

Proofreading is best done in Sigil's Book View, or in PageEdit, both allowing you to edit while reading, which is sadly not possible in ordinary e-ink e-readers.

After you have opened the EPUB file in Sigil, you also need to open the original PDF in the program you usually use for PDFs. That program can be Acrobat Reader or the freeware Sumatra (which I recommend) for Windows, Preview for Mac and Linux. Or, as in my illustrations here, dear old ABBYY.

You want to have the PDF open beside the EPUB in Sigil for quick access to the original PDF text as you proofread the EPUB in Sigil. Then you can quickly check the original if you aren't sure if there's a problem or not. In this case: did the text really say "habendi"? (Yes, it did.)

N.B. The highlights in the Sigil images are mine, so that the various examples will be clearly illustrated, while the PDF highlights show search results in ABBYY.



THORNDYKE was not a newspaper reader. He viewed with extreme disfavour all scrappy and miscellaneous forms of literature, which, by presenting a disorderly series of unrelated items of information, tended, as he considered, to destroy the habit of consecutive mental effort.

"It is most important," he once remarked to me, "habitually to pursue a definite train of thought, and to pursue it to a finish, instead of flitting indolently from one uncompleted topic to another, as the newspaper reader is so apt to do. Still, there is no harm in a daily paper—so long as you don't read it."

Accordingly, he patronised a morning paper, and his method of dealing with it was characteristic. The paper was laid on the table after breakfast, together with a blue pencil and a pair of office shears. A preliminary glance through the sheets enabled him to mark with the pencil those paragraphs that were to be read, and these were presently cut out and looked through, after which they were either thrown away or set aside to be pasted in an indexed book.

The whole proceeding occupied, on an average, a quarter of an hour.

On the morning of which I am now speaking he was thus engaged. The pencil had done its work, and the snick of the shears announced the final stage. Presently he paused with a newly-excised cutting between his fingers, and, after glancing at it for a moment, he handed it to me.

"Another art robbery," he remarked. "Mysterious affairs, these—as to motive, I mean. You can't melt down a picture or an ivory carving, and you can't put them on the market as they stand. The very qualities that give them their value make them totally unnegotiable."

"Yet I suppose," said I, "the really inveterate collector—the pottery or stamp maniac, for instance—will buy these contraband goods even though he dare not show them."

"Probably. No doubt the *cupiditas* *habendi*, the mere desire to possess, is the motive force rather than any intelligent purpose——"

The discussion was at this point interrupted by a knock at the door, and a moment later my colleague admitted two gentlemen. One of these I recognised as a Mr. Marchmont, a solicitor, for whom we had occasionally acted; the other was a stranger—a typical Hebrew of the blonde type—good-looking, faultlessly dressed, carrying a bandbox, and obviously in a state of the most extreme agitation.

"Good-morning to you, gentlemen," said Mr. Marchmont, shaking hands cordially. "I have brought a client of mine to see you, and when I tell you that his name is Solomon Löwe, it will be unnecessary for me to say what our business is."

"Oddly enough," replied Thorndyke, "we were, at the very moment when you knocked, discussing the bearings of his case."

"It is a horrible affair!" burst in Mr. Löwe. "I am distracted! I am ruined! I am in despair!"

He banged the bandbox down on the table, and flinging himself into a chair, buried his face in his hands.

"Come, come," remonstrated Marchmont, "we must be brave, we must be composed. Tell Dr. Thorndyke your story, and let us hear what he thinks of it."

He leaned back in his chair, and looked at his client with that air of patient fortitude that comes to us all so easily when we contemplate the misfortunes of other people.

"You must help us, sir," exclaimed Löwe, starting up again—"you must, indeed, or I shall go mad. But I shall tell you what has happened, and then you must act at once. Spare no effort and no expense. Money is no object—at least, not in reason," he added, with native caution. He sat down once more, and in perfect English, though with a slight German accent, proceeded volubly: "My brother Isaac is probably known to you by name."

Thorndyke nodded.

"He is a great collector, and to some extent a dealer—that is to say, he makes his hobby a profitable hobby."

"What does he collect?" asked Thorndyke.

"Everything," replied our visitor, flinging his hands apart with a comprehensive gesture—"everything that is precious and beautiful—pictures, ivories, jewels, watches, objects of art and *vertu*—everything. He is a Jew, and he has that passion for things that are rich and costly that has distinguished our race from the time of my namesake Solomon onwards. His house in Howard Street, Piccadilly, is at once a museum and an art gallery. The rooms are filled with cases of gems, of antique jewellery, of coins and historic relics—some of priceless value—and the walls are covered with paintings, every one of which is a

Find:

Replace:

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Regex | All HTML Files | Down

definite train of thought, and to pursue it to a finish, instead of flitting indolently from one uncompleted topic to another, as the newspaper reader is so apt to do. Still, there is no harm in a daily paper—so long as you don't read it."

Accordingly, he patronised a morning paper, and his method of dealing with it was characteristic. The paper was laid on the table after breakfast, together with a blue pencil and a pair of office shears. A preliminary glance through the sheets enabled him to mark with the pencil those paragraphs that were to be read, and these were presently cut out and looked through, after which they were either thrown away or set aside to be pasted in an indexed book.

The whole proceeding occupied, on an average, a quarter of an hour.

On the morning of which I am now speaking he was thus engaged. The pencil had done its work, and the snick of the shears announced the final stage. Presently he paused with a newly-excised cutting between his fingers, and, after glancing at it for a moment, he handed it to me.

"Another art robbery," he remarked. "Mysterious affairs, these—as to motive, I mean. You can't melt down a picture or an ivory carving, and you can't put them on the market as they stand. The very qualities that give them their value make them totally unnegotiable."

"Yet I suppose," said I, "the really inveterate collector—the pottery or stamp maniac, for instance—will buy these contraband goods even though he dare not show them."

"Probably. No doubt the *cupiditus* *habendi*, the mere desire to possess, is the motive force rather than any intelligent purpose——"

The discussion was at this point interrupted by a knock at the door, and a moment later my colleague admitted two gentlemen. One of these I recognised as a Mr. Marchmont, a solicitor, for whom we had occasionally acted; the other was a stranger—a typical Hebrew of the blonde type—good-looking, faultlessly dressed, carrying a bandbox, and obviously in a state of the most extreme agitation.

"Good-morning to you, gentlemen," said Mr. Marchmont, shaking hands cordially. "I have brought a client of mine to see you, and when I tell you that his name is Solomon Löwe, it will be unnecessary for me to say what our business is."

"Oddly enough," replied Thorndyke, "we were, at the very moment when you knocked, discussing the bearings of his case."

"It is a horrible affair!" burst in Mr. Löwe. "I am distracted! I am ruined! I am in despair!"

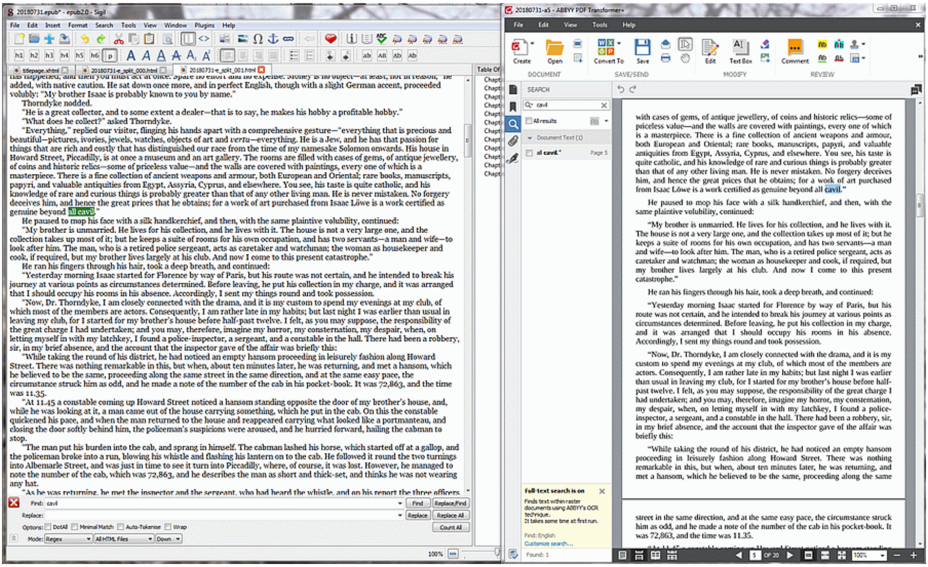
He banged the bandbox down on the table, and flinging himself into a chair, buried his face in his hands.

3 OF 20

Examples of possible problems:

OK, I didn't quickly find any really good examples, but let's say I couldn't figure out if this word really should be "all" or something else, and then what?

Searching for a common word like "all" will give you far too many results. In such cases, look for some uncommon word near the one you are really looking for. Here I chose "cavil":


20180731.epub* - epub2.0 - Sigil
File Edit Insert Format Search Tools View Window Plugins Help
titlepage.xhtml
20180731-e_split_000.html
20180731-e_split_001.html
Find
Replace/Find
Replace
Replace All
Count All
20180731-a5 - ABBYY PDF Transformer+
File Edit View Tools Help
Create
Open
Convert To
Save
Edit
Text Box
Comment
DOCUMENT
SAVE/SEND
MODIFY
REVIEW
SEARCH
All results
Document Text (1)
Page 5
100%

added, with native caution. He sat down once more, and in perfect English, though with a slight German accent, proceeded volubly: "My brother Isaac is probably known to you by name."

Thorndyke nodded.

"He is a great collector, and to some extent a dealer—that is to say, he makes his hobby a profitable hobby."

"What does he collect?" asked Thorndyke.

"Everything," replied our visitor, flinging his hands apart with a comprehensive gesture—"everything that is precious and beautiful—pictures, ivories, jewels, watches, objects of art and *vertu*—everything. He is a Jew, and he has that passion for things that are rich and costly that has distinguished our race from the time of my namesake Solomon onwards. His house in Howard Street, Piccadilly, is at once a museum and an art gallery. The rooms are filled with cases of gems, of antique jewellery, of coins and historic relics—some of priceless value—and the walls are covered with paintings, every one of which is a masterpiece. There is a fine collection of ancient weapons and armour, both European and Oriental; rare books, manuscripts, papyri, and valuable antiquities from Egypt, Assyria, Cyprus, and elsewhere. You see, his taste is quite catholic, and his knowledge of rare and curious things is probably greater than that of any other living man. He is never mistaken. No forgery deceives him, and hence the great prices that he obtains; for a work of art purchased from Isaac Löwe is a work certified as genuine beyond all cavil."

He paused to mop his face with a silk handkerchief, and then, with the same plaintive volubility, continued:

"My brother is unmarried. He lives for his collection, and he lives with it. The house is not a very large one, and the collection takes up most of it; but he keeps a suite of rooms for his own occupation, and has two servants—a man and wife—to look after him. The man, who is a retired police sergeant, acts as caretaker and watchman; the woman as housekeeper and cook, if required, but my brother lives largely at his club. And now I come to this present catastrophe."

He ran his fingers through his hair, took a deep breath, and continued:

"Yesterday morning Isaac started for Florence by way of Paris, but his route was not certain, and he intended to break his journey at various points as circumstances determined. Before leaving, he put his collection in my charge, and it was arranged that I should occupy his rooms in his absence. Accordingly, I sent my things round and took possession.

"Now, Dr. Thorndyke, I am closely connected with the drama, and it is my custom to spend my evenings at my club, of which most of the members are actors. Consequently, I am rather late in my habits; but last night I was earlier than usual in leaving my club, for I started for my brother's house before half-past twelve. I felt, as you may suppose, the responsibility of the great charge I had undertaken; and you may, therefore, imagine my horror, my consternation, my despair, when, on letting myself in with my latchkey, I found a police-inspector, a sergeant, and a constable in the hall. There had been a robbery, sir, in my brief absence, and the account that the inspector gave of the affair was briefly this:

"While taking the round of his district, he had noticed an empty hansom proceeding in leisurely fashion along Howard Street. There was nothing remarkable in this, but when, about ten minutes later, he was returning, and met a hansom, which he believed to be the same, proceeding along the same street in the same direction, and at the same easy pace, the circumstance struck him as odd, and he made a note of the number of the cab in his pocket-book. It was 72,863, and the time was 11.35.

"At 11.45 a constable coming up Howard Street noticed a hansom standing opposite the door of my brother's house, and, while he was looking at it, a man came out of the house carrying something, which he put in the cab. On this the constable quickened his pace, and when the man returned to the house and reappeared carrying what looked like a portmanteau, and closing the door softly behind him, the policeman's suspicions were aroused, and he hurried forward, hailing the cabman to stop.

"The man put his burden into the cab, and sprang in himself. The cabman lashed his horse, which started off at a gallop, and the policeman broke into a run, blowing his whistle and flashing his lantern on to the cab. He followed it round the two turnings into Albemarle Street, and was just in time to see it turn into Piccadilly, where, of course, it was lost. However, he managed to note the number of the cab, which was 72,863, and he describes the man as short and thick-set, and thinks he was not wearing any hat.

"As he was returning, he met the inspector and the sergeant, who had heard the whistle, and on his report the three officers

Find: cavil

Replace:

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Regex | All HTML Files | Down

cavil

al cavil."

with cases of gems, of antique jewellery, of coins and historic relics—some of priceless value—and the walls are covered with paintings, every one of which is a masterpiece. There is a fine collection of ancient weapons and armour, both European and Oriental; rare books, manuscripts, papyri, and valuable antiquities from Egypt, Assyria, Cyprus, and elsewhere. You see, his taste is quite catholic, and his knowledge of rare and curious things is probably greater than that of any other living man. He is never mistaken. No forgery deceives him, and hence the great prices that he obtains; for a work of art purchased from Isaac Löwe is a work certified as genuine beyond all cavil."

He paused to mop his face with a silk handkerchief, and then, with the same plaintive volubility, continued:

"My brother is unmarried. He lives for his collection, and he lives with it. The house is not a very large one, and the collection takes up most of it; but he keeps a suite of rooms for his own occupation, and has two servants—a man and wife—to look after him. The man, who is a retired police sergeant, acts as caretaker and watchman; the woman as housekeeper and cook, if required, but my brother lives largely at his club. And now I come to this present catastrophe."

He ran his fingers through his hair, took a deep breath, and continued:

"Yesterday morning Isaac started for Florence by way of Paris, but his route was not certain, and he intended to break his journey at various points as circumstances determined. Before leaving, he put his collection in my charge, and it was arranged that I should occupy his rooms in his absence. Accordingly, I sent my things round and took possession.

"Now, Dr. Thorndyke, I am closely connected with the drama, and it is my custom to spend my evenings at my club, of which most of the members are actors. Consequently, I am rather late in my habits; but last night I was earlier than usual in leaving my club, for I started for my brother's house before half-past twelve. I felt, as you may suppose, the responsibility of the great charge I had undertaken; and you may, therefore, imagine my horror, my consternation, my despair, when, on letting myself in with my latchkey, I found a police-inspector, a sergeant, and a constable in the hall. There had been a robbery, sir, in my brief absence, and the account that the inspector gave of the affair was briefly this:

"While taking the round of his district, he had noticed an empty hansom proceeding in leisurely fashion along Howard Street. There was nothing remarkable in this, but when, about ten minutes later, he was returning, and met a hansom, which he believed to be the same, proceeding along the same street in the same direction, and at the same easy pace, the circumstance struck him as odd, and he made a note of the number of the cab in his pocket-book. It was 72,863, and the time was 11.35.

Full-text search is on

Finds text within raster documents using ABBYY's OCR technique.
It takes some time at first run.

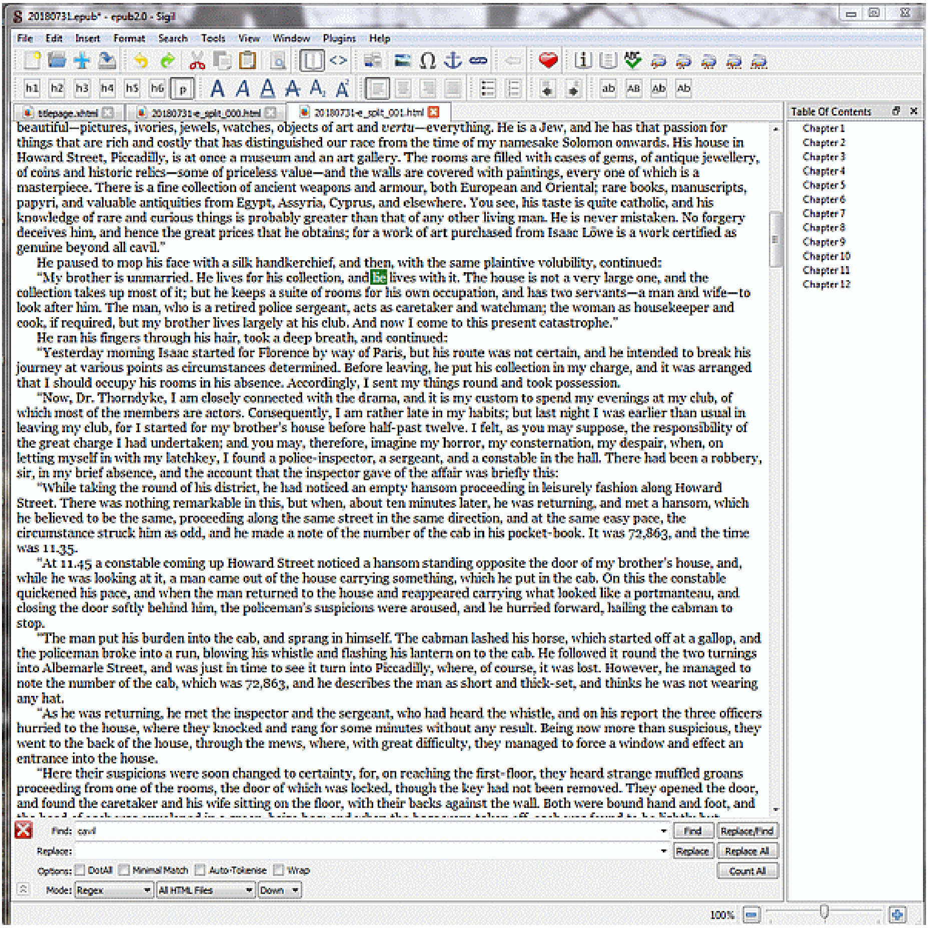
Find: English

Customize search...

Found: 1

5 OF 20

You will find few or no truly misspelled words in the book when a spellcheck has already been run, but there will likely be a number of misread “real” words, such as:



beautiful—pictures, ivories, jewels, watches, objects of art and *vertu*—everything. He is a Jew, and he has that passion for things that are rich and costly that has distinguished our race from the time of my namesake Solomon onwards. His house in Howard Street, Piccadilly, is at once a museum and an art gallery. The rooms are filled with cases of gems, of antique jewellery, of coins and historic relics—some of priceless value—and the walls are covered with paintings, every one of which is a masterpiece. There is a fine collection of ancient weapons and armour, both European and Oriental; rare books, manuscripts, papyri, and valuable antiquities from Egypt, Assyria, Cyprus, and elsewhere. You see, his taste is quite catholic, and his knowledge of rare and curious things is probably greater than that of any other living man. He is never mistaken. No forgery deceives him, and hence the great prices that he obtains; for a work of art purchased from Isaac Löwe is a work certified as genuine beyond all cavil.”

He paused to mop his face with a silk handkerchief, and then, with the same plaintive volubility, continued:

“My brother is unmarried. He lives for his collection, and lie lives with it. The house is not a very large one, and the collection takes up most of it; but he keeps a suite of rooms for his own occupation, and has two servants—a man and wife—to look after him. The man, who is a retired police sergeant, acts as caretaker and watchman; the woman as housekeeper and cook, if required, but my brother lives largely at his club. And now I come to this present catastrophe.”

He ran his fingers through his hair, took a deep breath, and continued:

“Yesterday morning Isaac started for Florence by way of Paris, but his route was not certain, and he intended to break his journey at various points as circumstances determined. Before leaving, he put his collection in my charge, and it was arranged that I should occupy his rooms in his absence. Accordingly, I sent my things round and took possession.

“Now, Dr. Thorndyke, I am closely connected with the drama, and it is my custom to spend my evenings at my club, of which most of the members are actors. Consequently, I am rather late in my habits; but last night I was earlier than usual in leaving my club, for I started for my brother's house before half-past twelve. I felt, as you may suppose, the responsibility of the great charge I had undertaken; and you may, therefore, imagine my horror, my consternation, my despair, when, on letting myself in with my latchkey, I found a police-inspector, a sergeant, and a constable in the hall. There had been a robbery, sir, in my brief absence, and the account that the inspector gave of the affair was briefly this:

“While taking the round of his district, he had noticed an empty hansom proceeding in leisurely fashion along Howard Street. There was nothing remarkable in this, but when, about ten minutes later, he was returning, and met a hansom, which he believed to be the same, proceeding along the same street in the same direction, and at the same easy pace, the circumstance struck him as odd, and he made a note of the number of the cab in his pocket-book. It was 72,863, and the time was 11.35.

“At 11.45 a constable coming up Howard Street noticed a hansom standing opposite the door of my brother's house, and, while he was looking at it, a man came out of the house carrying something, which he put in the cab. On this the constable quickened his pace, and when the man returned to the house and reappeared carrying what looked like a portmanteau, and closing the door softly behind him, the policeman's suspicions were aroused, and he hurried forward, hailing the cabman to stop.

“The man put his burden into the cab, and sprang in himself. The cabman lashed his horse, which started off at a gallop, and the policeman broke into a run, blowing his whistle and flashing his lantern on to the cab. He followed it round the two turnings into Albemarle Street, and was just in time to see it turn into Piccadilly, where, of course, it was lost. However, he managed to note the number of the cab, which was 72,863, and he describes the man as short and thick-set, and thinks he was not wearing any hat.

“As he was returning, he met the inspector and the sergeant, who had heard the whistle, and on his report the three officers hurried to the house, where they knocked and rang for some minutes without any result. Being now more than suspicious, they went to the back of the house, through the mews, where, with great difficulty, they managed to force a window and effect an entrance into the house.

“Here their suspicions were soon changed to certainty, for, on reaching the first-floor, they heard strange muffled groans proceeding from one of the rooms, the door of which was locked, though the key had not been removed. They opened the door, and found the caretaker and his wife sitting on the floor, with their backs against the wall. Both were bound hand and foot, and

Find: cavil

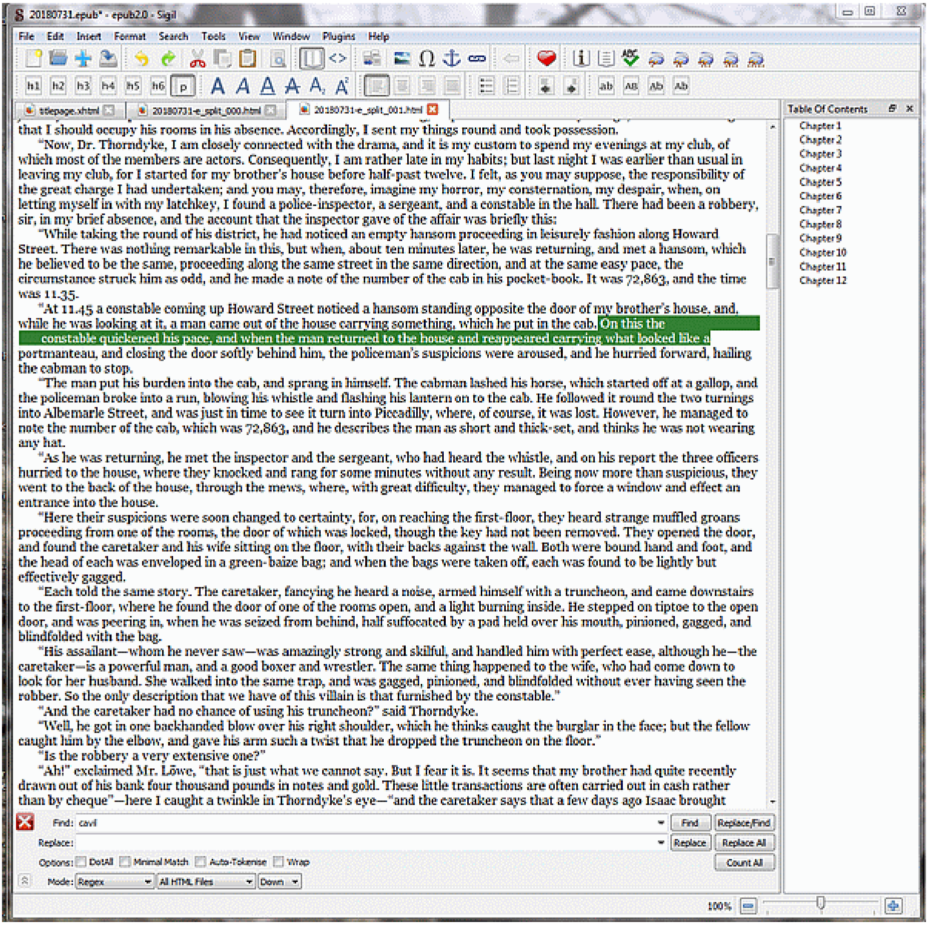
Replace:

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Regex ▾ All HTML Files ▾ Down ▾

In cases like this one you don’t need to check the original. It is obvious that you should change “lie” to “he” (which is actually such a common OCR error that you might want to run a search for it before proofreading).

In addition, there may be some formatting errors left in spite of previous automated checks, notably bad line breaks (a new line where there shouldn’t be one) and the opposite: missing line breaks. If in doubt, search the PDF for an unusual word and check the original before editing.



that I should occupy his rooms in his absence. Accordingly, I sent my things round and took possession.

"Now, Dr. Thorndyke, I am closely connected with the drama, and it is my custom to spend my evenings at my club, of which most of the members are actors. Consequently, I am rather late in my habits; but last night I was earlier than usual in leaving my club, for I started for my brother's house before half-past twelve. I felt, as you may suppose, the responsibility of the great charge I had undertaken; and you may, therefore, imagine my horror, my consternation, my despair, when, on letting myself in with my latchkey, I found a police-inspector, a sergeant, and a constable in the hall. There had been a robbery, sir, in my brief absence, and the account that the inspector gave of the affair was briefly this:

"While taking the round of his district, he had noticed an empty hansom proceeding in leisurely fashion along Howard Street. There was nothing remarkable in this, but when, about ten minutes later, he was returning, and met a hansom, which he believed to be the same, proceeding along the same street in the same direction, and at the same easy pace, the circumstance struck him as odd, and he made a note of the number of the cab in his pocket-book. It was 72,863, and the time was 11.35.

"At 11.45 a constable coming up Howard Street noticed a hansom standing opposite the door of my brother's house, and, while he was looking at it, a man came out of the house carrying something, which he put in the cab. On this the constable quickened his pace, and when the man returned to the house and reappeared carrying what looked like a portmanteau, and closing the door softly behind him, the policeman's suspicions were aroused, and he hurried forward, hailing the cabman to stop.

"The man put his burden into the cab, and sprang in himself. The cabman lashed his horse, which started off at a gallop, and the policeman broke into a run, blowing his whistle and flashing his lantern on to the cab. He followed it round the two turnings into Albemarle Street, and was just in time to see it turn into Piccadilly, where, of course, it was lost. However, he managed to note the number of the cab, which was 72,863, and he describes the man as short and thick-set, and thinks he was not wearing any hat.

"As he was returning, he met the inspector and the sergeant, who had heard the whistle, and on his report the three officers hurried to the house, where they knocked and rang for some minutes without any result. Being now more than suspicious, they went to the back of the house, through the mews, where, with great difficulty, they managed to force a window and effect an entrance into the house.

"Here their suspicions were soon changed to certainty, for, on reaching the first-floor, they heard strange muffled groans proceeding from one of the rooms, the door of which was locked, though the key had not been removed. They opened the door, and found the caretaker and his wife sitting on the floor, with their backs against the wall. Both were bound hand and foot, and the head of each was enveloped in a green-baize bag; and when the bags were taken off, each was found to be lightly but effectively gagged.

"Each told the same story. The caretaker, fancying he heard a noise, armed himself with a truncheon, and came downstairs to the first-floor, where he found the door of one of the rooms open, and a light burning inside. He stepped on tiptoe to the open door, and was peering in, when he was seized from behind, half suffocated by a pad held over his mouth, pinioned, gagged, and blindfolded with the bag.

"His assailant—whom he never saw—was amazingly strong and skilful, and handled him with perfect ease, although he—the caretaker—is a powerful man, and a good boxer and wrestler. The same thing happened to the wife, who had come down to look for her husband. She walked into the same trap, and was gagged, pinioned, and blindfolded without ever having seen the robber. So the only description that we have of this villain is that furnished by the constable."

"And the caretaker had no chance of using his truncheon?" said Thorndyke.

"Well, he got in one backhanded blow over his right shoulder, which he thinks caught the burglar in the face; but the fellow caught him by the elbow, and gave his arm such a twist that he dropped the truncheon on the floor."

"Is the robbery a very extensive one?"

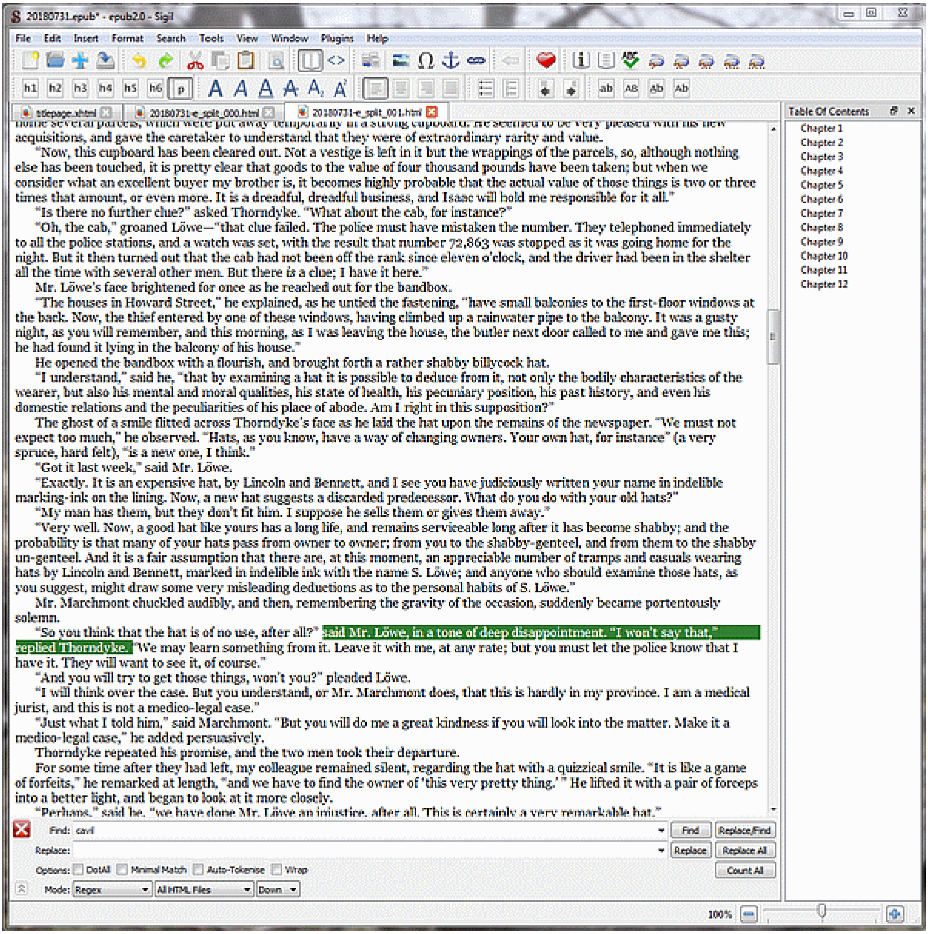
"Ah!" exclaimed Mr. Lōwe, "that is just what we cannot say. But I fear it is. It seems that my brother had quite recently drawn out of his bank four thousand pounds in notes and gold. These little transactions are often carried out in cash rather than by cheque"—here I caught a twinkle in Thorndyke's eye—"and the caretaker says that a few days ago Isaac brought

Find: cavl

Replace:

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Regex All HTML Files Down



home several parcels, which were put away temporarily in a strong cupboard. He seemed to be very pleased with his new acquisitions, and gave the caretaker to understand that they were of extraordinary rarity and value.

"Now, this cupboard has been cleared out. Not a vestige is left in it but the wrappings of the parcels, so, although nothing else has been touched, it is pretty clear that goods to the value of four thousand pounds have been taken; but when we consider what an excellent buyer my brother is, it becomes highly probable that the actual value of those things is two or three times that amount, or even more. It is a dreadful, dreadful business, and Isaac will hold me responsible for it all."

"Is there no further clue?" asked Thorndyke. "What about the cab, for instance?"

"Oh, the cab," groaned Löwe—"that clue failed. The police must have mistaken the number. They telephoned immediately to all the police stations, and a watch was set, with the result that number 72,863 was stopped as it was going home for the night. But it then turned out that the cab had not been off the rank since eleven o'clock, and the driver had been in the shelter all the time with several other men. But there *is* a clue; I have it here."

Mr. Löwe's face brightened for once as he reached out for the bandbox.

"The houses in Howard Street," he explained, as he untied the fastening, "have small balconies to the first-floor windows at the back. Now, the thief entered by one of these windows, having climbed up a rainwater pipe to the balcony. It was a gusty night, as you will remember, and this morning, as I was leaving the house, the butler next door called to me and gave me this; he had found it lying in the balcony of his house."

He opened the bandbox with a flourish, and brought forth a rather shabby billycock hat.

"I understand," said he, "that by examining a hat it is possible to deduce from it, not only the bodily characteristics of the wearer, but also his mental and moral qualities, his state of health, his pecuniary position, his past history, and even his domestic relations and the peculiarities of his place of abode. Am I right in this supposition?"

The ghost of a smile flitted across Thorndyke's face as he laid the hat upon the remains of the newspaper. "We must not expect too much," he observed. "Hats, as you know, have a way of changing owners. Your own hat, for instance" (a very spruce, hard felt), "is a new one, I think."

"Got it last week," said Mr. Löwe.

"Exactly. It is an expensive hat, by Lincoln and Bennett, and I see you have judiciously written your name in indelible marking-ink on the lining. Now, a new hat suggests a discarded predecessor. What do you do with your old hats?"

"My man has them, but they don't fit him. I suppose he sells them or gives them away."

"Very well. Now, a good hat like yours has a long life, and remains serviceable long after it has become shabby; and the probability is that many of your hats pass from owner to owner; from you to the shabby-genteel, and from them to the shabby un-genteel. And it is a fair assumption that there are, at this moment, an appreciable number of tramps and casuals wearing hats by Lincoln and Bennett, marked in indelible ink with the name S. Löwe; and anyone who should examine those hats, as you suggest, might draw some very misleading deductions as to the personal habits of S. Löwe."

Mr. Marchmont chuckled audibly, and then, remembering the gravity of the occasion, suddenly became portentously solemn.

"So you think that the hat is of no use, after all?" said Mr. Löwe, in a tone of deep disappointment. "I won't say that," replied Thorndyke. "We may learn something from it. Leave it with me, at any rate; but you must let the police know that I have it. They will want to see it, of course."

"And you will try to get those things, won't you?" pleaded Löwe.

"I will think over the case. But you understand, or Mr. Marchmont does, that this is hardly in my province. I am a medical jurist, and this is not a medico-legal case."

"Just what I told him," said Marchmont. "But you will do me a great kindness if you will look into the matter. Make it a medico-legal case," he added persuasively.

Thorndyke repeated his promise, and the two men took their departure.

For some time after they had left, my colleague remained silent, regarding the hat with a quizzical smile. "It is like a game of forfeits," he remarked at length, "and we have to find the owner of 'this very pretty thing.' " He lifted it with a pair of forceps into a better light, and began to look at it more closely.

"Perhaps," said he, "we have done Mr. Löwe an injustice, after all. This is certainly a very remarkable hat."

Find: cavil

Replace:

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Regex All HTML Files Down

Remember to press Ctrl-S (or Cmd-S on Mac) after each correction, so that you don't lose your work if something should happen.

Check with e-reader (BV & CV)

One final step before you can sit back and be really pleased with what you have achieved…

Load your EPUB into your e-reader and make sure that you like what you see. Also, test it with one or more e-readers on your computer.

And then I say: **WELL DONE!** Welcome to the club of proud creators of quality EPUBs!

Appendix 1: Searches for Easy Copying

This is a list of the correction/fix searches for easy copying - note that some of them may wreck your EPUB if run with the “Replace All” option, so you do need to read the comments in the Corrections and Fixes chapter before use. Also, the numbering below may refer to alternatives or order to run, so you need to check up on that, too.

```
SEMANTIC TAGS

Adding bodymatter when front- and backmatter have been adde
Find
<body>
Replace with:
<body epub:type="bodymatter">


Adding chapter sections when other bodymatter sections have
Find
<body epub:type="bodymatter">
Replace with:
<body epub:type="bodymatter">\n<section epub:type="chapter
Before saving:
Find
</body>
Replace with:
</section>\n</body>


REGEX CORRECTIONS & FIXES:

Missing chapter heading tags 1
Find:
<p>Chapter (.*?)</p>
Replace with:
<h2>Chapter \1</h2>


Missing chapter heading tags 2
Find:
<body>\s+<p class="(.*?)">(\d+)</p>
```

Replace with:

```
<body><h2>\2</h2>
```

Missing chapter heading tags 3

Find:

```
<body>\s+<p>(\d+)</p>
```

Replace with:

```
<body><h2>\1</h2>
```

Page numbers remaining after OCR 1

Find:

```
<p class="(.*?)">Page \d+</p>
```

Replace with:

nothing

Page numbers remaining after OCR 2

Find:

```
<p class="(.*?)">\d+</p>
```

Replace with:

nothing

Page numbers remaining after OCR 3

Find:

```
<p>Page \d+</p>
```

Replace with:

nothing

Page numbers remaining after OCR 4

Find:

```
<p>\d+</p>
```

Replace with:

nothing

Bad line breaks 1a

Find:

```
</p>\s+<p class="(.*?)">([a-z]\w*\s)
```

Replace with:

```
 \2
```

N.B. the space before \2

Bad line breaks 1b

Find:

```
</p>\s+<p>([a-z]\w*\s)
```

Replace with:

```
 \1
```

N.B. the space before \1

Bad line breaks 2a

Find:

```
([^.?!;:>])</p>\s+<p class="(.*?)">
```

Replace with:

```
\1 
```

N.B. the space after \1

Bad line breaks 2b

Find:

```
([^.?!;:>])</p>\s+<p>
```

Replace with:

```
\1 
```

N.B. the space after \1

Missing line breaks in conversations - dq

Find:

```
”(.)“
```

Replace with:

```
”</p>\n\n<p class="(.*?)">“
```

or

```
”</p>\n\n<p>“
```

Missing line breaks in conversations - sq

Find:

```
’(.)‘
```

Replace with:

```
’</p>\n\n<p class="(.*?)">‘
```

or

```
’</p>\n\n<p>‘
```

Removing incorrect line breaks due to leftover hyphens 1
Find:
```
([a-z])-</p>\s+<p class="(.*?)">([a-z])
```
Replace with:
```
\1\2
```

Removing incorrect line breaks due to leftover hyphens 2
Find:
```
([a-z])-</p>\s+<p>([a-z])
```
Replace with:
```
\1\2
```

Removing <sup> tags
Find:
```
<sup>(.*?)</sup>
```
Replace with:
```
\1 (or nothing)
```

Removing <sub> tags
Find:
```
<sub>(.*?)</sub>
```
Replace with:
```
\1 (or nothing)
```

Remove anchors
Find:
```
<a id=(.*?)></a>
```
Replace with:
nothing (blank "Replace" box)

Remove hyperlinks
Find:
```
<a href=(.*?)>(.*?)</a>
```
Replace with:
```
\2
```

Remove mistaken underlining

```
Find:
<span style="text-decoration:underline;">(.*?)</span>
Replace with:
\1


ell instead of exclamation mark 1
Find:
[l](\s[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)]
!\1


ell instead of exclamation mark 2
Find:
[l]</p>
Replace with:
!</p>


ell instead of exclamation mark dq
Find:
[l]”</p>
Replace with:
!”</p>
ell instead of exclamation mark sq
Find:
[l]’</p>
Replace with:
!’</p>
P instead of question mark 1
Find:
[P]([\s][A-Z"])
Replace with:
?\1


P instead of question mark 2
Find:
[P]</p>
Replace with:
?</p>


P instead of question mark dq
```

Find:

```
[P]”</p>
```

Replace with:

```
?”</p>
```

P instead of question mark sq

Find:

```
[P]’</p>
```

Replace with:

```
?’</p>
```

Bold tag applied to single letter

```
<b>(.?)</b>
```

Replace with:

```
\1
```

Italic tag applied to single letter

```
<i>(.?)</i>
```

Replace with:

```
\1
```

For a different first line following h2 headings

Find:

```
</h2>\s+<p>
```

Replace with:

```
</h2><p class="firstline">
```

For a different first line following h3 headings

Find:

```
</h3>\s+<p>
```

Replace with:

```
</h3><p class="firstline">
```

For a different first line following scenebreaks/scenechan

Find:

```
<p class="scenebreak">(.*?)</p>\s+<P class="(.*?)">
```

Replace with:

```
<p class="scenebreak">\1</p>\n\n+< class="firstline">
```

```
NORMAL MODE CORRECTIONS & FIXES:

Replace non-breaking space with blank space 1
Find:
&nbsp;
Replace with:
a blank space


Replace non-breaking space with blank space 2
Find:
&#160;
Replace with:
a blank space


Search for bum for burn, etc.
Search for lie for he, if it seems to happen a lot in the
Search for Pie or Fie for He or She if it seems to happen

one instead of I - no. 1
Find:
 1 
Replace with:
 I 
N.B. spaces before and after


one instead of I - no. 2
Find:
<p class="(.*?)">1
Replace with:
<p class="(.*?)">I


one instead of I - no. 3
Find:
<p>1
Replace with:
<p>I
```

```
ell instead of I - no. 1
Find:
 l
Replace with:
 I
N.B. spaces before and after


CASE SENSITIVE:
ell instead of I - no. 2
Find:
<p class="(.*?)">l
Replace with:
<p class="(.*?)">I


CASE SENSITIVE:
ell instead of I - no. 3
Find:
<p>1
Replace with:
<p>I


Double main quotefix, left side
Find:
“‘
or
“ ‘
or
“&#160;‘
Replace with:
<span class="quotefix">“&#160;</span>‘


Double main quotefix, right side
Find:
’”
or
’ ”
or
’&#160;”
Replace with:
<span class="quotefix">’&#160;</span>”
```

```
Single main quotefix, left side:
Find:
‘“
or
‘ “
or
‘&#160;“
Replace with:
<span class="quotefix">‘&#160;</span>“


Single main quotefix, right side:
Find:
”’
or
” ’
or
”&#160;’
Replace with:
<span class="quotefix">”&#160;</span>’
```

Appendix 2: Various Recognized EPUB Types

...presented in the order they might have in a novel.

Of course, not all of them will be used in all novels, so pick the appropriate ones for your book. This way, the book will be clearly structured from the screen reader’s point of view.

I have found no standard EPUB types for features such as book lists or text copied from the back of the dust jacket. I would suggest using "appendix" for these “extras”, or just make them unnamed sections.

```
<body epub:type="frontmatter">
<section epub:type="cover">
(...)
</section>
</body>
```

```
<body epub:type="frontmatter">
<section epub:type="halftitlepage">
(...)
</section>
</body>
```

```
<body epub:type="frontmatter">
<section epub:type="titlepage">
(...)
</section>
</body>
```

```
<body epub:type="frontmatter">
<section epub:type="copyright-page">
(...)
</section>
</body>
```

```
<body epub:type="frontmatter">
<section epub:type="acknowledgments">
(...)
</section>
</body>
```

```
< body epub:type = "frontmatter" >
< section epub:type = "credits" >
(...)
< /section >
< /body >
```

Acknowledgments or credits? If the paper book doesn’t indicate which, choose yourself.

```
< body epub:type = "frontmatter" >
< section epub:type = "dedication" >
(...)
< /section >
< /body >
```

```
< body epub:type = "bodymatter" >
< section epub:type = "epigraph" >
(...)
< /section >
< /body >
```

```
< body epub:type = "bodymatter" >
< section epub:type = "foreword" >
(...)
< /section >
< /body >
```

```
< body epub:type = "bodymatter" >
< section epub:type = "introduction" >
(...)
< /section >
< /body >
```

```
< body epub:type = "bodymatter" >
< section epub:type = "preamble" >
(...)
< /section >
< /body >
```

```
< body epub:type = "bodymatter" >
< section epub:type = "preface" >
```

```
(...)
</section>
</body>
```

Please don’t ask me to explain the difference between preamble and preface, because I really don’t know. They might as well be called “introduction” for all I know…

```
<body epub:type="bodymatter">
<section epub:type="prologue">
<h2>Prologue</h2>
(...)
</section>
</body>
```

```
<body epub:type="bodymatter">
<section epub:type="part">
<h1>Part I</h1>
(...)
</section>
</body>
```

or

```
<body epub:type="bodymatter">
<section epub:type="part">
<h1>Part I</h1>
(...)
<section epub:type="chapter">
<h2>Chapter 1</h2>
(...)
</section>
<section epub:type="chapter">
<h2>Chapter 2</h2>
(...)
</section>
</section>
</body>
```

```
<body epub:type="bodymatter">
<section epub:type="chapter">
<h2>Chapter 1</h2>
(...)
</section>
</body>
```

```
<body epub:type="bodymatter">
<section epub:type="epilogue">
<h2>Epilogue</h2>
(...)
</section>
</body>
```

```
<body epub:type="backmatter">
<section epub:type="appendix">
(...)
</section>
</body>
```

```
<body epub:type="backmatter">
<section epub:type="colophon">
(...)
</section>
</body>
```

Appendix 3: Table of Basic HTML Entities and Codes

EPUB3 is stricter than EPUB2 when it comes to accepting, or rather not accepting, HTML entities that do not have the same counterparts in XML. That is the major part of them. Only three of the character entities commonly used in novels are accepted in XML: &amp; (&), &lt; (<), and &gt; (>), along with the less commonly used &quot; (') and &apos; (") for single and double straight quote marks. That’s it.

You need to use XML entities, often called HTML codes, instead. It follows that you need to learn some codes instead of entities if you work in Code View and occasionally need to make sure that you get the correct character. So here is a short list; I have included the items I most commonly have written as entities. N.B. changing to codes will not break backwards compatibility with EPUB2. The non-XML entities shouldn’t really have been accepted in XHTML files in the past either.

> *To be sure, there are entities/codes for all kinds of characters, but most of the time you just press the right key on your keyboard, and the character appears as expected both in Book View and Code View. The entities/codes below are the commonly used ones not commonly found on keyboards.*

Character	HTML Entity	HTML Code
©	&copy;	&#169;
—	&mdash;	&#8212;
‘	&lsquo;	&#8216;
’	&rsquo;	&#8217;
“	&ldquo;	&#8220;
”	&rdquo;	&#8221;
non-breaking space	&nbsp;	&#160;